Balanced Trustworthiness, Safety, and Privacy in Vehicle-to-Vehicle Communications

Qianhong Wu, Josep Domingo-Ferrer, *Senior Member, IEEE*, and Úrsula González-Nicolás

***Abstract*—Vehicular ad hoc networks (VANETs) are being designed to improve traffic safety and efficiency. To meet this goal, the messages disseminated in VANETs must be trustworthy. We propose a privacy-preserving system that guarantees message trustworthiness in vehicle-to-vehicle (V2V) communications. Vehicle privacy is provided as long as a vehicle does not attempt to endorse the same message more than once. In spite of a message having been validly endorsed, if it is later found to be false, the system offers the possibility of *a posteriori* tracing the message generator and its endorsers. Our proposal demonstrates a number of distinctive features. The system is equipped with both *a priori* and *a posteriori* countermeasures. The threshold used for *a priori* endorsement can adaptively change according to the message urgency and traffic context, rather than being preset in the system design stage as in existing schemes. The verification of authenticated V2V messages is accelerated by batch message-processing techniques. Simulation results illustrate that the system maintains its performance under various traffic conditions.**

***Index Terms*—Group signature, information security, network security, protocol design, vehicular communication.**

I. Introduction

With computer networks spreading into a variety of new environments, vehicle-to-vehicle (V2V) communications can be expected to be available by 2011 [1]. The IEEE 802.11p task group is working on the dedicated short-range communication (DSRC) standard to support wireless communications for vehicles and roadside infrastructures [2]. Car manufacturers and telecommunication industries are gearing up to equip each car with onboard units (OBUs) that allow vehicles to communicate with each other, as well as to supply roadside units (RSUs). The OBUs and RSUs form a vehicular ad hoc network (VANET), by which vehicles can disseminate messages to other vehicles in their vicinity. This mechanism can be exploited to improve safety, traffic efficiency, driver assistance, and transportation regulation. However, malicious vehicles can also leverage this mechanism to send fraudulent messages for their own profit or just to jeopardize the traffic system. Hence, it is essential to design a system to ensure that the transmission comes from a trusted source and has not been tampered with.

Driving privacy or vehicle anonymity is another critical concern in VANETs. As noted in [3], a lot can be inferred on the driver's privacy if the whereabouts and the driving pattern of a car can be tracked. It is possible for attackers to trace vehicles by using cameras or physical tracking. However, such physical attacks can only trace specific targets and are much more expensive than attacks based on monitoring vehicular communications. Hence, attention is devoted to guaranteeing privacy against the latter attacks. Furthermore, anonymity can potentially degrade the trustworthiness of V2V communications because the generators of messages are indistinguishable and cannot be identified for liability. Therefore, the trustworthiness, safety, and privacy of V2V communications need to be well balanced.

A conventional transportation-regulation system (without VANETs) may involve vehicle manufacturers, a transportation regulation office, the traffic police, and judges. As commonly suggested [4], [5], it is reasonable to assume that those conventional entities have their corresponding electronic counterparts in a VANET. Such centralized authorities can be responsible for enrolling legitimate vehicles, validating their identities, and issuing electronic certificates to vehicles. They will take care of performing regular (e.g., annual) health checks of vehicles, collecting vehicular communications, reconstructing accidents, tracing drivers, optimizing traffic, and relieving congestion. Furthermore, these administration units make centralized security infrastructures such as public-key infrastructures (PKIs) that are usable in VANETs. Unlike the nodes of other types of mobile ad hoc networks (MANETs), which are limited in power and computation, the computation devices embedded in vehicles can be expected to have substantial computational capacity, storage space, and power supply. These operational features of VANETs comprise of the physical basis on which indispensable yet better security can be provided in VANETs.

A. Related Work

VANETs can improve traffic safety only if the messages sent by vehicles are trustworthy. Dealing with fraudulent messages

Manuscript received March 21, 2009; revised July 17, 2009 and September 16, 2009. First published October 20, 2009; current version published February 19, 2010. This work was supported in part by the Spanish Government under Project TIN2009-11689 "RIPUP," Project TSI2007-65406-C03-01 "E-AEGIS," and Project CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES," by the Government of Catalonia under Grant 2009 SGR 1135, and by the Chinese National Science Foundation under Project 60970115 and Project 60970116. The work of J. Domingo-Ferrer was supported in part by the Government of Catalonia under a Catalan Institution for Research and Advanced Studies Acadèmia Research Prize. The review of this paper was coordinated by Prof. Y. Zhang.

Q. Wu is with the United Nations Educational, Scientific, and Cultural Organization, Chair in Data Privacy, Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, 43007 Tarragona, Spain, and also with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Computer, Wuhan University, Wuhan 430072, China (e-mail: qianhong.wu@urv.cat).

J. Domingo-Ferrer and Ú. González-Nicolás are with the United Nations Educational, Scientific, and Cultural Organization (UNESCO) Chair in Data Privacy, Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, 43007 Tarragona, Spain (e-mail: josep.domingo@urv.cat; ursula.gonzaleznicolas@urv.cat).

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

Digital Object Identifier 10.1109/TVT.2009.2034669

is a thorny issue for safety engineers due to the self-organized operation of VANETs. The situation is further deteriorated by the privacy requirements, since the message generators, i.e., the vehicles, are anonymous and cannot be identified when acting maliciously. A number of schemes have been proposed to reduce fraudulent messages; such proposals fall into two classes, namely, *a posteriori* and *a priori*.

With *a posteriori* countermeasures, punitive action will be taken against vehicles who have been proven to have originated fraudulent messages. Cryptographic authentication technologies have extensively been exploited to offer *a posteriori* countermeasures. Some proposals use regular digital signatures [6]–[8] to enable tracing malicious vehicles. To make this approach work, a PKI is suggested in VANETs [9], [10]. When multiple vehicles observe the same driving environment, to endorse the generated message, they need to authenticate the same or a similar message. This raises the issue of authentication of aggregated data. In [11], the authors propose ways to authenticate identical messages. Another way to deal with authentication of aggregated data is suggested in [12]. This proposal can handle messages that are similar but not identical and expects nodes receiving multiple messages with similar information to summarize the information in them using only syntactic aggregation.

A critical issue posed by vehicular message authentication is driver privacy. Since the public keys used to verify the authenticated messages are linked to specific users, attackers can trace vehicles by observing vehicular communications. Hence, mechanisms must be adopted to guarantee vehicle/driver privacy when vehicles authenticate messages. Along this research line, there are two main approaches: pseudonymous mechanisms and group signatures. In a pseudonymous mechanism, the certificate authorities produce multiple pseudonyms for each vehicle so that attackers cannot trace the vehicles producing the signatures in different periods under different pseudonyms, except if the certificate authorities open the identities of the vehicles. The IEEE 1609.2 Draft Standard [13] proposes the distribution of short-lived certificates to enable vehicle privacy. In [14], the authors propose to use a set of anonymous keys that frequently change (e.g., every couple of minutes), depending on the driving speed. Each key can be used only once, and it expires after its usage; only one key can be used at a time. The authors of [15] propose to use a silent period to hamper linkability between pseudonyms or, alternatively, to create groups of vehicles and restrict vehicles in one group from hearing the messages of other groups.

This conditional anonymity of pseudonymous authentication will help determine the liability of drivers in case of accidents. The downside of this approach is the necessity for the generation, delivery, storage, and verification of numerous certificates for all the keys. To mitigate this heavy overhead, [16] presents an approach to enable vehicle OBUs to generate their own pseudonyms without interacting with the certificate authority. The mechanism is realized with the help of group signatures. In [4], a novel group-signature-based security framework is proposed, which relies on tamper-resistant devices (requiring password access) to prevent adversarial attacks on vehicular networks. However, they provide no concrete instantiation or experimental analysis. In [5], the authors propose a security and privacy-preserving protocol for VANETs by integrating the techniques of group signatures and identity-based signatures. With their approach, the heavy load of pseudonym management can be eliminated, which makes their scheme very efficient.

A priori countermeasures attempt to prevent the generation of fraudulent messages. VANETs can improve traffic safety and efficiency only if vehicular messages are correct. The application of information-theoretic measures to anomaly detection was previously studied in the literature [17]–[19] but mainly in the context of the wired Internet. Most notably, [19] successfully applied the notion of relative entropy (also known as the Kullback–Leibler distance) to measure the similarity between two data sets. A general proposal that handles both the detection and correction of malicious data is given in [20] by assuming that the simplest explanation of some inconsistency in the received information is most probably the correct one.

Observing the heavy overhead incurred by the aforementioned protocols to correct erroneous messages, some new proposals suggest more efficient threshold mechanisms [21]–[23] to achieve a similar goal; a message is trusted only if it was endorsed by a number of vehicles in the vicinity. This approach is based on the assumption that most users are honest and will not endorse any message containing false data. Another implicit assumption is the common sense that the more the people that endorse a message, the more trustworthy it is. Among these schemes, the proposals in [23] may be the most efficient while enabling anonymity of message originators by exploiting secret sharing techniques. However, 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" [24].

We observe that neither *a posteriori* nor *a priori* countermeasures are by their own sufficient to secure VANETs. By taking strict punitive action, *a posteriori* countermeasures can exclude some rational attackers, but they are ineffective against irrational attackers such as terrorists. Even for rational attackers, damage has already occurred when punitive action is taken. As for *a priori* countermeasures, although the underlying assumption that there is a majority of honest vehicles in VANETs generally holds, it cannot be excluded that a number of malicious vehicles (e.g., controlled by a criminal organization) greater than or equal to the threshold are present in specific locations. Furthermore, for convenience in implementation, existing schemes use an even stronger assumption that the number of honest vehicles in all cases should be at least a preset threshold. However, such a universally valid threshold does not exist in practice. Indeed, the threshold should somehow take the traffic density and the message scope into account: A low density of vehicles calls for a lower threshold, whereas a high density and a message relevant to all of the traffic in a city require a sufficiently high threshold.

The situation is aggravated by the anonymity technologies used in some proposals. A system preserves anonymity when it does not require the identity of its users to be disclosed. Without anonymity, attackers can trace all the vehicles by monitoring the communication in VANETs. However, anonymity can also

weaken *a posteriori* and *a priori* countermeasures. Indeed, due to anonymity, attackers can send fraudulent messages without fear of being caught, and as a result, no punitive action can be taken against them. Furthermore, some proposals provide strong anonymity, i.e., unlinkability. Unlinkability implies that a verifier cannot distinguish whether two signatures come from the same vehicle or two vehicles. This feature may enable malicious vehicles to weaken *a priori* countermeasures by mounting the so-called Sybil attack: A vehicle generates a fraudulent message and then endorses the message by computing on it as many signatures as required by the threshold in use; since signatures are unlinkable, no one can find out that all of them come from the same vehicle. These observations present the challenging issue of balancing trustworthiness, safety, and privacy in VANET V2V communications.

B. Our Contribution

Bearing in mind that enhancing safety and traffic efficiency is one of the main thrusts behind VANETs, we propose an efficient system for balancing public safety and vehicle privacy. To this end, we present a new primitive called the message-linkable group signature (MLGS), in which a vehicle stays anonymous if it produces one signature on each message. However, if it produces two signatures on one message, then the attacker will be found by a trusted authority, which effectively thwarts the Sybil attack in a privacy-preserving system. This novel technology also enables us to realize a threshold-adaptive authentication in which the threshold can adaptively change in light of the message context. In [25], we presented a general framework by using the aforementioned ideas, but no concrete scheme was implemented. In this paper, we implement a new scheme that is efficient in both communication and computation. Experiments also demonstrate that the proposal can achieve high resilience to threats against the trustworthiness of V2V communications and vehicle privacy without introducing any significant performance penalty.

Compared with previous privacy-preserving proposals for secure vehicular communications, our proposal illustrates a number of unique features: 1) Both *a priori* and *a posteriori* countermeasures are used to thwart attackers. To the best of our knowledge, ours is the first system equipped with both types of countermeasures. In our scheme, vehicles only trust the messages endorsed by a number of anonymous vehicles greater than or equal to a threshold, and the anonymous vehicles endorsing cheating messages can later be traced. When a vehicle receives multiple signatures on the same message, it can distinguish by itself whether the message was signed by the same cheating vehicle multiple times or by multiple honest vehicles. Although a trusted authority is required to trace the cheating vehicle, the trusted authority does not need to always be online because the receiving vehicle needs to contact the trusted authority only if some judicial procedure is invoked. 2) The threshold in our proposal can change according to the traffic context, unlike most previous schemes in which the threshold has to be preset during the stage of system design. This feature enables our proposal to be deployed in complicated traffic settings. We also note that a recent proposal in [26] allows the threshold to be changed in different scenarios. 3) Finally, a provably secure batch-verification method is presented to accelerate the validation of messages, which has been a bottleneck in previous schemes. With the new verification method, a batch of messages can be verified as if they were a single one. The provable security guarantees that our batch verification approach will not degrade the security of the system, even if a very large number of messages are verified in a single batch. Since vehicles are suggested to periodically send messages over a single hop every 300 ms [10], vehicles may receive a large number of messages to be validated in a very short time interval. Hence, our fast verification approach is critical in making authentication implementable in VANETs.

The remainder of this paper is organized as follows. Section II outlines a general security framework of VANETs. The proposed scheme is detailed in Section III. Techniques to speed up message verification are presented in Section IV. Simulations are reported in Section V to evaluate the performance of the scheme, followed by a conclusion in Section VI.

II. New Security Framework for VANETs

A. Security Requirements

Since the main security threats in VANETs are violations of public safety and vehicle privacy, an attacker is defined to be an entity who wants to successfully cheat honest vehicles by diffusing false information or compromise the privacy of honest vehicles by monitoring the communications in VANETs. Accordingly, to obtain an implementable system to enhance the trustworthiness in privacy-preserving V2V communications, we consider the following three types of security requirements.

1) *Threshold authentication*. A message is viewed as trustworthy only after it has been endorsed by at least t vehicles, where t is a threshold. The threshold mechanism is an *a priori* countermeasure that improves the confidence of other vehicles in a message. In itself, the threshold does not stop malicious behaviors but makes them more difficult to materialize. Furthermore, the authentication may provide arguments if such behaviors occur and must later be judged.
2) *Anonymity*. There is anonymity if, by monitoring the communication in a VANET, message originators cannot be identified, except perhaps by designated parties. The goal is to protect the privacy of vehicles. Since message authentication requires knowledge of a public identity such as a public key or the license plate, if no anonymity mechanism was provided, an attacker could easily trace any vehicle by monitoring the VANET communication. This would be undesirable for the drivers, and the anonymity mechanism is intended to disable this type of attack.
3) *Revocability*. Revocability means that, if necessary, designated parties can identify the originator and the endorsers of any doubtable message. The goal here is to balance personal privacy and public safety. If anonymity is realized without any revocability mechanism, an attacker can anonymously broadcast authenticated wrong

messages to fool other vehicles without fear of being caught, which may seriously compromise public safety. The revocability mechanism is an *a posteriori* countermeasure intended to fight this impunity situation.

B. Framework Based on MLGSs

Group signatures have been investigated for many years [27]. In a group-signature scheme, each group member can anonymously sign messages on behalf of the group, and a group manager can open the identity of the author of any group signature if necessary. Group signatures are useful for securing VANETs, but they may be subject to the Sybil attack because of unlinkability. A linkable group signature [28] is a variant of group signatures. In a linkable group signature, it is easy to distinguish the group signatures produced by the same signer, even though the signer is anonymous. Linkable group signatures can thwart the Sybil attack but are not compatible with vehicle privacy due to the linkability of signer identities, i.e., the various message endorsements signed by a certain vehicle can be linked. To fill this gap, we explore a more elegant notion of linkability in group signatures and present a new primitive referred to as the MLGS, in which, given two signatures on *the same message*, one can easily decide whether the two signatures are anonymously generated by the same group member or not.

Definition 1 (MLGS): An MLGS is an interactive protocol between a register manager, a tracing manager, a set of group members, and a set of verifiers. It consists of the following polynomial-time algorithms.

1) `Setup`: On input of a security parameter λ, this algorithm outputs the public system parameter denoted by π, including a description of the system.
2) `KGen`: On input of the system parameter π, this algorithm outputs the public–private key pairs of the register manager, the tracing manager, and group members.
3) `Join`: It is an interactive protocol between group members, the register manager, and the tracing manager. The outputs of a group member, the register manager, and the tracing manager are, respectively, a group certificate, a list of registered group members, and some secret tracing information to trace group signatures.
4) `GSign`: On input of the system parameter π, a message m, a private group member key, and the corresponding group certificate, this algorithm outputs a group signature σ of m.
5) `GVerify`: On input of the system parameter π, a message m, a signature σ, and the public key of the register manager, this algorithm outputs a bit 1 or 0 to represent that σ is valid or not.
6) `GTrace`: On input of the system parameter π, a message m, a *valid* group signature σ, and the secret tracing information, the tracing manager outputs the identity of the author of σ.

A secure MLGS scheme must be correct, unforgeable, anonymous, traceable, and message linkable. These properties are defined as follows.

1) `Correctness`. This states that `GVerify` always outputs 1 if all parties honestly follow the MLGS scheme.
2) `Unforgeability`. This states that any polynomial-time user who has not registered to the group has only a negligible probability of producing a valid group signature.
3) `Anonymity`. An MLGS scheme is anonymous if, given a valid message–signature pair from one of two group members, any polynomial-time attacker has only a probability of $0.5 + \varepsilon$ of guessing the correct originator of the message–signature pair, where ε is negligible.
4) `Traceability`. An MLGS scheme is traceable if any polynomial-time attacker has only a negligible probability of producing a valid group signature such that the output of `GTrace` is not the identity of the group signature originator.
5) `Message linkability`. An MLGS scheme is message linkable if there exists a polynomial-time algorithm that, on input of a message m and valid group signatures σ_1 and σ_2 on m, outputs a bit 1 or 0 to represent whether σ_1 and σ_2 are generated by the same author or not.

Based on MLGSs, we propose a general framework for threshold authentication with revocable anonymity in VANETs. In this framework, each vehicle registers to a vehicle administration office serving as a group registration manager. When t vehicles wish to endorse some message, they can independently generate an MLGS on that message. A verifying vehicle trusts the message after validating t MLGSs on it. If later the message is found to be incorrect, the police and judges can together trace the t cheating signers.

From the security properties of MLGS schemes, it is clear that the aforementioned framework satisfies the required properties of threshold-variable authentication, anonymity, and revocability in VANETs. If $t-1$ vehicles produce t signatures on the same message, then there exists a group member who has been involved in generating at least two signatures. Such an impersonation can easily be identified since the MLGS scheme is message linkable.

III. Threshold V2V Authentication Protocol

We propose a protocol to promote the trustworthiness of privacy-preserving vehicle-generated messages. Underlying this is an efficient MLGS scheme that is constructed in a modular way.

A. Computational Assumptions

Our scheme is realized in bilinear pairing groups [29], [30]. Let `PGen` be an algorithm that, on input of a security parameter 1^λ, outputs a tuple $\Upsilon = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e)$, where finite cyclic groups $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ have the same prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$ is an efficient nondegenerate bilinear map such that $e(g_1, g_2) \neq 1$ and, for all $h_1 \in \mathbb{G}_1$, $h_2 \in \mathbb{G}_2$, and $u, v \in \mathbb{Z}$, $e(h_1^u, h_2^v) = e(h_1, h_2)^{uv}$.

Our scheme relies on several well-studied computational assumptions, i.e., the decisional Diffie–Hellman (DDH) assumption and the Diffie–Hellman knowledge (DHK) assumption [31]. Let p be a λ-bit prime, $\mathbb{G}$ be a finite cyclic group of prime order p, and g be the generator of $\mathbb{G}$. Those two assumptions are briefly reviewed next.

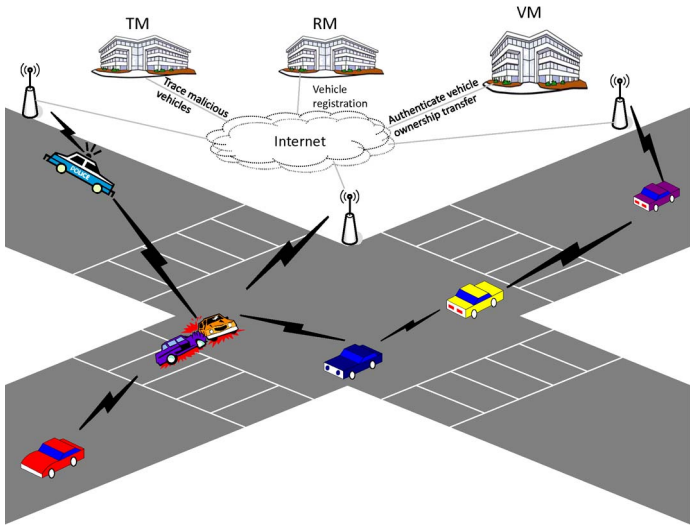

Fig. 1. System architecture.

Definition 2 (DDH Assumption): Given $(g, g^a, g^b, g^c) \in \mathbb{G}^4$, where $a, b, c \in \mathbb{Z}_p^*$, for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the probability of determining whether $c = ab$ is negligibly away from 1/2.

With a proper implementation, the DDH holds in $\mathbb{G}_1$ (see [30]).

Definition 3 (DHK Assumption [31]): Given $(g, g^x) \in \mathbb{G}^2$ for randomly chosen $x \in \mathbb{Z}_p^*$, any PPT adversary $\mathcal{A}$ has only a negligible probability of creating a Diffie–Hellman tuple (g, g^x, g^r, g^{xr}) without knowing r.

We will exploit the DHK assumption in pairing groups Υ. That is, given that $g_1 \in \mathbb{G}_1$ and $(g_2, g_2^x) \in \mathbb{G}_2^2$, where $x \in \mathbb{Z}_p^*$ is randomly chosen, any PPT adversary $\mathcal{A}$ has only a negligible probability to create an accompanied Diffie–Hellman tuple $(g_2, g_2^x, g_1^r, g_1^{xr})$ without knowing r.

B. System Architecture

The system consists of four parties, i.e., the vehicles, the vehicle manufacturer, the registration manager, and the group tracing manager. They are denoted by $\mathcal{V}$, $\mathcal{VM}$, $\mathcal{RM}$, and $\mathcal{TM}$, respectively. When a vehicle is sold, $\mathcal{VM}$ and $\mathcal{V}$ sign a contract to clarify that the ownership of the vehicle has legally been transferred. With the signed contract, the vehicle can register to $\mathcal{RM}$ to become a legal group member. During registration, some tracing information will securely be sent to $\mathcal{TM}$ so that $\mathcal{TM}$ can trace the vehicle if it later maliciously behaves. $\mathcal{RM}$ can be implemented by transportation companies for commercial vehicles or by vehicle management agencies for private cars. $\mathcal{TM}$ can be implemented by police offices. The trust assumptions among the parties are similar to those in the real world. We assume that $\mathcal{VM}$, $\mathcal{RM}$, and $\mathcal{TM}$ honestly behave and are semitrusted: They are not allowed access to the private keys of the vehicles. The vehicles might be malicious but rational in the sense that they will attack the system for their own benefit only if they do not expect to be caught. The system architecture is illustrated in Fig. 1.

TABLE I
NOTATIONS AND THEIR MEANINGS

Notation: Meaning
$\mathcal{A}$: A Probabilistic Polynomial-Time (PPT) attacker
π: The set of public system parameters
$\mathbb{G}_i (i = 1, 2, 3)$: Finite cyclic group of prime order p
g_i: a random generator of $\mathbb{G}_i$
$U_2, h_2 \in \mathbb{G}_2$: Public system parameters
ϕ: An isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$
$U_1 = \phi(U_2)$: Public system parameter
$h_1 = \phi(h_2)$: Public system parameter
$H_1()$: A cryptographic hash function from $\{0,1\}^*$ to $\mathbb{G}_1$
$\mathcal{RM}$: The registration manager
$H()$: A cryptographic hash function from $\{0,1\}^*$ to $\mathbb{Z}_p$
(A, Z): The public-private key pair of $\mathcal{RM}$
$\mathcal{VM}$: The vehicle manufacturer
$\mathcal{TM}$: The tracing manufacturer
$\mathcal{V}$: A vehicle
(Y, y): The public-private key pair of $\mathcal{V}$
$K_{\mathcal{V}} = (K_1, K_2)$: The group certificate of $\mathcal{V}$
$T = g_2^y$: The tracing information of $\mathcal{V}$
m: A message
σ: A signature on message m
$M = (m, \sigma)$: A message appended with a signature
σ_i: The i-th component of σ
N: The number of signatures on n messages
n: The number of messages in a batch

C. System Setup

The involved semitrusted parties can set up the system as specified below. No private input is required during the generation of the system parameters. Let $\Upsilon = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e) \leftarrow \mathtt{PGen}(1^\lambda)$ be generated as aforementioned, where the DDH and DHK assumptions hold in $\mathbb{G}_1$. Assume that ϕ is a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ such that $\phi(g_2) = g_1$. Let h_2 and U_2 be randomly chosen from $\mathbb{G}_2$ [30]. The system parameters are $\pi = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e; h_2, h_1; U_2, U_1; H_1, H \rangle$. The system parameters can be embedded into OBUs. In such a way, the system is initialized. Table I summarizes the notations used hereafter in this paper.

D. Key Generation

$\mathcal{RM}$ randomly selects Z from $\mathbb{G}_1$ and computes $A = e(Z, g_2)$. The public–private key pair of $\mathcal{RM}$ is (A, Z), where the public key is available to all entities, while the private key is known only to $\mathcal{RM}$. Furthermore, $\mathcal{TM}$ generates a public/secret key pair to receive escrowed identities of the vehicles. Each vehicle generates its public key $Y = U_1^y$ for a random value $y \in \mathbb{Z}_p^*$. The vehicle's secret key is y. A vehicle generates a new key pair when ownership changes. The generated public key must be signed by the previous owner or the vehicle manufacturer when selling the vehicle. For simplicity, we assume in the sequel that the vehicle has just been sold by its manufacturer.

E. Vehicle Registration

To register to a VANET, a vehicle $\mathcal{V}$ contacts $\mathcal{RM}$ and $\mathcal{TM}$ using a confidential channel and goes through the following steps.

1) A vehicle first contacts the tracing manager via a confidential channel to show that the vehicle has legitimately been obtained. It also sends some trapdoor information of its public key such that the tracing manager can trace the vehicle later if necessary. Then, the tracing manager issues a signature for the vehicle to convince the registration manager that the vehicle's identity has successfully been escrowed. To this end, the vehicle $\mathcal{V}$ computes $T = g_2^y$ and sends (Y, T) to $\mathcal{TM}$, as well as the signature by $\mathcal{VM}$ on Y. $\mathcal{TM}$ checks that
   a) the signature by $\mathcal{VM}$ on Y is valid to guarantee that the vehicle has legitimately been obtained;
   b) $e(Y, g_2) = e(U_1, T)$ to guarantee that the vehicle can be traced if it is maliciously behaving.

   If both checks hold, $\mathcal{TM}$ generates a signature on Y and sends it to $\mathcal{V}$. $\mathcal{TM}$ saves (Y, T) and the signature by $\mathcal{VM}$ on Y into its local database. After this procedure, the legal authorities serving as a group-tracing manager have gathered evidence on the vehicle ownership transfer chains and tracing information T.
2) After escrowing its identity, the vehicle $\mathcal{V}$ registers to the registration manager with its public key and the signature of its public key by the tracing manager. If $\mathcal{V}$ successfully registers, then it will receive a signature on its public key from the registration manager. This signature serves as a group certificate, and $\mathcal{V}$ can use this certificate and its private key to generate signatures on any message. The detailed registration procedure is described as follows.
   a) $\mathcal{V}$ submits Y and the signature by $\mathcal{TM}$ to $\mathcal{RM}$. $\mathcal{V}$ runs with $\mathcal{RM}$ a zero-knowledge proof protocol denoted by $ZK\{y|Y = U_1^y\}$ [2] to prove that $\mathcal{V}$ knows secret y such that $Y = U_1^y$ without leaking the information of y. Then, $\mathcal{RM}$ checks that the signatures by $\mathcal{TM}$ and $ZK\{y|Y = U_1^y\}$ are valid. This is to guarantee that $\mathcal{V}$ has successfully escrowed its public key Y to the tracing manager and is the owner of Y. If both checks hold, $\mathcal{RM}$ generates a signature on Y by randomly choosing $k \leftarrow \mathbb{Z}_p^*$ and computing $K_1 = g_1^k$ and $K_2 = Z(h_1 Y)^{-k}$.
   b) $\mathcal{RM}$ returns $K_\mathcal{V} = (K_1, K_2)$ to $\mathcal{V}$. $\mathcal{V}$ computes K_1^y and checks that $e(K_2, g_2)e(K_1, h_2)e(K_1^y, U_2) = A$ to validate the signature. If the check holds, $\mathcal{V}$ successfully registers to the VANET and can use $K_\mathcal{V}$ as a group certificate and its secret key to sign any message.

TABLE II
FORMAT OF VEHICLE-GENERATED MESSAGES (LENGTH IN BYTES)

Message ID	Payload	Timestamp	TTL	Group ID	Signature
2	100	4	1	2	128

F. Anonymous Threshold Authentication

A vehicle-generated message consists of six fields: message ID, payload, timestamp, time to live (TTL), group ID, and signature. The message ID defines the message type, and the payload field may include the information on the vehicle's position, direction, speed, traffic events, event time, and so on. According to [2], the payload of a message is 100 B. A timestamp specifies the signature generation time, which is used to prevent replay attacks. It also ensures that an honest vehicle can report the same traffic situation at different times without being accused of multiple signatures on the same message. The TTL field determines how long the message is allowed to remain in the VANET. The group ID is used to identify which group the vehicle belongs to. The signature field is the vehicle's signature on the first five fields. We denote the first five fields by m and all of the six fields by M. The timestamp and TTL fields imply that the system requires time synchronization. Since VANETs are assumed to be deployed with centralized authorities, it is not difficult to realize time synchronization. Table II specifies the suggested length for each field.

Slightly unlike the message format suggested in [5], we additionally require the TTL field to be signed. If the TTL field is not signed, a malicious vehicle can revive an outdated message and mount a corpse attack by merely modifying the TTL field of ever-valid messages.[1] Hence, it is preferable to let the TTL field be signed to improve security.

To endorse a message $m \in \mathbb{Z}_p^*$, $\mathcal{V}$ generates an MLGS on the message and then sends it to other vehicles. The group signature consists of three parts. The first part is a randomization of the group certificate to show that the signer is a group member while keeping the signer anonymous. The second part is a randomization of the group member's public key and a message-link identifier. For the same message, $\mathcal{V}$ can only produce one message-link identifier. The last part is a regular signature signed with the secret key hidden in the randomized member public key and the message-link identifier. Vehicles from different groups can endorse the same message. Since different vehicles can independently endorse the same message, the system performance will be affected in the case that vehicles

[1] It is clear from the context when a message means either the first five fields or all the six fields. We do not stress it hereafter.

from different groups endorse the same message. The signing procedure consists of the following computations.

1) *Randomize the group certificate*. Randomly select $s \leftarrow \mathbb{Z}_p^*$. Compute $\sigma_1 = K_1 g_1^s$ and $\sigma_2 = K_2(h_1 Y)^{-s}$.
2) *Randomize the member public key*. Compute $\sigma_3 = \sigma_1^y$. Note that y is $\mathcal{V}$'s secret key.
3) *Produce the message-link identifier*. Compute $\sigma_4 = H_1(m)^y$. Note that $\mathcal{V}$ can only generate one identifier for the same message.
4) *Generate a signature using the secret member key*. With this procedure, the signer, i.e., the endorsing vehicle, proves the knowledge of y hidden in $\sigma_3 = \sigma_1^y$ and $\sigma_4 = H_1(m)^y$. It is the standard transformation [32] from a zero-knowledge proof to a signature under a one-time public key (σ_3, σ_4). It consists of the following computations by the signer.
   a) Randomly select $r \leftarrow \mathbb{Z}_p^*$.
   b) Compute commitments $R_1 = H_1(m)^r$ and $R_2 = \sigma_1^r$.
   c) Obtain challenge σ_5 from the aforementioned values: $\sigma_5 = H(m\|\sigma_1\|\sigma_2\|\sigma_3\|\sigma_4\|R_1\|R_2)$.
   d) Answer the challenge with $\sigma_6 = r - \sigma_5 y \mod p$.
5) *Output the group signature*. Output $\sigma = (\sigma_1, \ldots, \sigma_6)$ as a group signature of m.
6) *Formulate the message*. Form the vehicle-generated message according to the format specified in Table II and broadcast it to vehicles in the vicinity.

G. Verification

When receiving a message, a vehicle discards the obsolete messages by checking the TTL field. The verifying vehicle also discards messages containing a value of σ_4, which has appeared in previous messages (as explained earlier, the repetition of σ_4 indicates that the same message is being signed more than once by the same anonymous signer). After a vehicle receives t nondiscarded group signatures σ on the same message m, it does the following for each signature.

1) Check $e(\sigma_2, g_2)e(\sigma_1, h_2)e(\sigma_3, U_2) = A$ to validate the group certificate in a blind manner. This procedure uses the group manager's public key A and system parameters g_2, h_2, and U_2 to verify that (σ_1, σ_2) is a signature on the vehicle's randomized public key σ_3.
2) Check $\sigma_5 = H(m\|\sigma_1\|\sigma_2\|\sigma_3\|\sigma_4\|H_1(m)^{\sigma_6}\sigma_4^{\sigma_5}\|\sigma_1^{\sigma_6}\sigma_3^{\sigma_5})$ to validate the signature. This verification convinces the verifier with the standard arguments that the vehicle $\mathcal{V}$ knows a secret value y satisfying both $\sigma_3 = \sigma_1^y$ and $\sigma_4 = H_1(m)^y$.

If both checks hold for all t signatures, the vehicle considers the message to be valid. If any verification fails, the vehicle discards the message as invalid (nonauthenticated). Note that the threshold t can adaptively be changed according to the type of message. For instance, if the message is an alert about an emergency braking by the vehicle ahead, the threshold can be set as low as 1. However, if the message is an announcement that will affect many vehicles, the threshold can be set to be appropriately high to improve the trustworthiness by also taking into account the vehicle density in the vicinity of the message source.

TABLE III
PERFORMANCE COMPARISON SUMMARY

	Message size	Sign	Verify
Ours	237 Bytes	6 Exps	1 Pairing + 6 Exps
[5]	301 Bytes	9 Exps + 1 Pairing	1 Pairing + 6 Exps

H. Trace Doubtable Messages

If a vehicle produced a group signature σ on the message m and this message was found to be fraudulent, a membership tracing operation is started to determine the real identity of the signature originator. The tracing manager first checks the validity of the signature σ on m and then looks up its local database to find (T, Y) such that $e(\sigma_3, g_2) = e(\sigma_1, T)$. If a match is found, then $\mathcal{TM}$ outputs Y as the identity of the original signer. Otherwise, it outputs an error message. According to [33], a pairing computation takes about 3.6 ms, and it takes about 1 h to trace a malicious vehicle in a group of up to 1 million vehicles. Such a tracing time is bearable and can be accelerated by tracing in parallel or dividing the vehicles in multiple groups, also noting that the tracing procedure will seldom be invoked in practice.

I. Performance Analysis

The factors that dominate the performance for securing vehicular messages are the message length and the overhead of message generation and verification. The length of vehicle-generated messages can be expressed as $L_M = L_{\mathrm{m-ID}} + L_{\mathrm{p-load}} + L_{\mathrm{t-stamp}} + L_{\mathrm{TTL}} + L_{\mathrm{g-ID}} + L_{\mathrm{sig}}$.

To provide a typical security level of 2^{80}, we can set p to be a 170-bit-long prime, and then, the element in $\mathbb{G}_1$ is 171 bits long [30], and $L_{\mathrm{sig}} = 128$ B. Thus, from Table II, $L_M = 2 + 100 + 4 + 1 + 2 + 128 = 237$ B. As for the computation, the signing procedure in our protocol requires six exponentiations, and the verification needs one pairing. Here, we do not distinguish a multibase exponentiation (pairing) from a single-base exponentiation (pairing), as they take similar time [34]. Table III summarizes the comparison between our scheme and the up-to-date scheme [5] for $t = 1$, as their scheme does not support a threshold mechanism. From Table III, our proposal is comparable with [5] in this case.

J. Security Analysis

The properties of MLGSs allow any t vehicles to distributively endorse any message without leaking their identities. The threshold t can change according to the message and traffic context. Any verifier can check whether a message is endorsed by t or more vehicles. A vehicle endorsing the same message more than once can also be detected; the tracing manager can revoke an anonymous vehicle endorsing a message that later turns out to be incorrect. To meet the aforementioned secure requirements, we show that the underlying MLGS scheme is correct, unforgeable, anonymous, traceable, and message

linkable. These properties are analyzed as follows. The proofs of Claims 1–3 are given in the Appendix.

Claim 1: The aforementioned scheme is correct.

This claim implies that the messages generated by vehicles honestly following the protocol will always be accepted. Hence, these messages can be used to guide other vehicles and potentially improve traffic safety and efficiency.

Claim 2: An attacker cannot forge a message to cheat honest vehicles.

Unforgeability guarantees that, if a vehicle does not register to the VANET, it cannot generate messages accepted by other vehicles, even if the cheating vehicle is allowed to access valid messages over the VANET. If a message passes the verification procedure, it must be an intact fresh message generated by a registered vehicle. This implies that the attacker cannot cheat other vehicles by forging a new valid message, modifying an existing valid message, or replaying a once valid but now expired message.

Claim 3: The originators of valid messages are anonymous.

This property means that an attacker cannot distinguish messages from the various vehicles in a VANET. More specifically, given one valid message and two vehicles in a VANET, an attacker cannot decide the originator of the message with a probability non-negligibly greater than 1/2. Hence, an attacker cannot trace the vehicles by monitoring the communication in the VANET, and the identity privacy of vehicles is protected.

Claim 4: The trusted third party (i.e., the tracing manager) can trace the anonymous generator of any valid message, and any honest vehicle can detect a message authenticated twice by the same vehicle.

Proof: Due to the unforgeability of the aforementioned group signature, the part of the signature under a one-time public key shows that $\sigma_3 = \sigma_1^y$ and $\sigma_4 = H_1(m)^y$, where y is the secret key of some group member (if y was not the secret key of any member, this would indicate a successful forgery and contradict the unforgeability). Hence, with the tracing information $T = g_2^y$, the tracing manager can trace the signer by checking that $e(\sigma_3, g_2) = e(\sigma_1, T)$ (see Section III-H). This property enables the legal authorities to identify and trace malicious messages, which is a deterrent for cheating vehicles. If the same signer signs the same message twice, then the two signatures share the same component $\sigma_4 = H_1(m)^y$. Hence, the signer can trivially be linked by comparing two signatures of the same message. ■

This fact guarantees that cheating vehicles can always be identified. If a cheating vehicle endorses a wrong message, it can be traced by a third party. If a vehicle tends to cheat other vehicles by endorsing the same message more than once with multiple signatures, then other vehicles can easily link the multiple signatures to the same vehicle. This kind of message can be either simply discarded or sent to the third party to trace the cheating vehicle. Hence, the Sybil attack can be avoided in our privacy-preserving scheme.

IV. Speedup Message Verification in VANETs

Although our scheme is very efficient in both communication and computation, it is very important to further speed up message verification because each vehicle periodically receives a large number of messages for verification.

A. Batch-Verification Lemma

In what follows, we explore some new techniques derived from [35] to enable batch verification of messages in our authentication protocol.

Lemma 1 (Batch-Verification Lemma): To verify the following exponentiation equations for $i = 1, \ldots, n$:

$$g_i^{x_i} h_i^{y_i} = 1 \tag{1}$$

where $x_i, y_i \in \mathbb{Z}_p^*$ are known, and g_i and h_i are two elements of a finite cyclic group $\mathbb{G}$ of prime order p, we randomly pick a vector $\Delta = (r_1, \ldots, r_n)$ for $r_i \in \{0,1\}^l$ and verify that

$$\prod_{i=1}^{n} g_i^{r_i x_i} h_i^{r_i y_i} = 1. \tag{2}$$

We accept (1) if (2) holds. Then, a batch $\{(g_i, h_i) | i = 1, \ldots, n\}$ will always be accepted if it is valid, while an invalid batch will be accepted with a probability of at most 2^{-l}.

The aforementioned lemma can naturally be extended to batch verifications of bilinear equations. In this case, we only need to additionally note that $1 = e(g_1, g_2)^a e(h_1, g_2)^b$ can equivalently be rewritten as $1 = e(g_1^a h_1^b, g_2)$ to save computation due to the bilinearity and the fact that exponentiations in $\mathbb{G}_1$ are more efficient than those in $\mathbb{G}_3$.

B. Batch Signature Verification

For the sake of better understanding the following fast verification method, we slightly modify the signing procedure by adding $R_1 = H_1(m)^r$ and $R_2 = \sigma_1^r$ to the signature (see Section III-F). That is, the resulting group signature is now $\sigma = (\sigma_1, \ldots, \sigma_6, R_1, R_2)$. Clearly, this modification does not affect any security property of the group signature because R_1 and R_2 can be reconstructed from $\sigma_1, \sigma_3, \sigma_4, \sigma_5$, and σ_6 [see the following verification equations in (4)]. We equivalently rewrite the verification equations as follows:

$$e(\sigma_2, g_2) e(\sigma_1, h_2) e(\sigma_3, U_2) = A \tag{3}$$

$$R_2 = \sigma_1^{\sigma_6} \sigma_3^{\sigma_5} \quad R_1 = H_1(m)^{\sigma_6} \sigma_4^{\sigma_5} \tag{4}$$

$$\sigma_5 = H\left(m \| \sigma_1 \| \sigma_2 \| \sigma_3 \| \sigma_4 \| R_1 \| R_2\right). \tag{5}$$

The dominant overhead in the verification of the VANET messages is due to group signature verification. Observe that a multibase pairing computation (respectively, multibase exponentiation) has almost the same overhead as a single-base pairing (respectively, single-base exponentiation) [34]. A potentially fast verification method is to verify the signatures in batch rather than to verify them one by one. This is possible because each vehicle periodically receives a large number of messages to be verified.

Now, we move to the batch-signature-verification algorithm. Let $(S_1, \ldots, S_N)$ be the N group signatures to be verified for n distinct messages $\{m'_1, \ldots, m'_n\}$, where $N \geq n$. To simplify

the notation, let $\{m'_1, \ldots, m'_n\} = \{m_1, \ldots, m_N\}$. Since $N \geq n$, it is possible that $m_i = m_j$ for $i \neq j$. Without loss of generality, we assume that the message m_j has signature S_j. As aforementioned, $S_j = (\sigma_{j,1}, \ldots, \sigma_{j,6}, R_{j,1}, R_{j,2})$. The signatures can come from different groups and be under group public keys $A_1, \ldots, A_N$, where $A_i = A_j$ is also possible for $i \neq j$.

Suppose that l is a positive integer. To simultaneously verify the N signatures, by exploiting Lemma 1, we randomly select $a_j, b_j, c_j \in \{0,1\}^l$ and check that

$$e\left(\prod_{j=1}^{N} \sigma_{j,2}^{a_j}, g_2\right) e\left(\prod_{j=1}^{N} \sigma_{j,1}^{a_j}, h_2\right) e\left(\prod_{j=1}^{N} \sigma_{j,3}^{a_j}, U_2\right) = \prod_{j=1}^{N} A_j^{a_j} \tag{6}$$

$$1 = \prod_{j=1}^{N} R_{j,1}^{-b_j} R_{j,2}^{-c_j} H_1(m_j)^{b_j \sigma_{j,6}} \sigma_{j,4}^{b_j \sigma_{j,5}} \sigma_{j,1}^{c_j \sigma_{j,6}} \sigma_{j,3}^{c_j \sigma_{j,5}} \tag{7}$$

$$\sigma_{j,5} = H\left(m_j \| \sigma_{j,1} \| \sigma_{j,2} \| \sigma_{j,3} \| \sigma_{j,4} \| R_{j,1} \| R_{j,2}\right) (1 \leq j \leq N). \tag{8}$$

If all checks hold, the batch of messages are accepted; otherwise, they are rejected. Note that, although we allow messages from different groups to be processed in the same batch, we assume that these groups share the public system parameters π (see Section III-C). If the system parameters are different, the batch verification can also similarly be employed, but this is much less efficient. Hence, we suggest that different groups share the same system parameters, which will also simplify the system deployment. For the soundness of the aforementioned batch verifications, we state the following claim.

Claim 5: In the aforementioned batch verification, a valid batch will always be accepted, while an invalid batch will be accepted with a probability of at most 2^{-2l}.

Proof: By exploiting Lemma 1, the verification given by (7) holds with a probability of at most 2^{-l} if the equations in (4) are invalid. Similarly, the verification given by (6) also holds with a probability of at most 2^{-l} if the equations in (3) are invalid. These two verifications are independent, and the verification given by (8) does not use batch techniques. Hence, the proposed batch verification accepts an invalid batch with a probability of at most $2^{-l} \cdot 2^{-l} = 2^{-2l}$. ■

The probability of accepting an invalid batch can be made negligible by properly setting parameter l. For instance, setting $l = 32$ yields a probability of accepting an invalid batch as low as 2^{-64}, which is affordable for most applications. We also notice that, if there is one invalid message, then the n messages in the batch will be rejected. For an individual message, the rejection probability under the batch approach is $1 - \rho^n$, where ρ is the probability that a message is valid; this is certainly higher than the rejection probability of $1 - \rho$ of individual verification, but we next argue that such an increase in the rejection probability is not a serious toll. In practice, there are only very few invalid messages due to the assumption of an honest vehicle majority in VANETs. Furthermore, the invalid signatures can efficiently be identified using a binary search method [36], [37] on a batch rejected as invalid. As shown in [38], batch verification is preferable to the naive individual verification, even if bad signatures must later be searched, and the proportion of bad signatures exceeds 10% of the total batch size.

We grossly compare the overhead of individual message verification with that of batch techniques for normal messages. For N messages, without using the batch approach, we need N multibase pairing computations, $2N$ multibase exponentiations, and N hashes. However, in the aforementioned batch verification of N messages, we need only one multibase pairing computation, five multibase exponentiations, and N hashes. According to state-of-the-art experimental results in [38], a typical pairing takes ten times longer than one exponentiation in $\mathbb{G}_1$, and compared with an exponentiation, the overhead of hash computation is negligible. Hence, the batch approach offers an about N-fold cost reduction. Note that the heavy overhead of message verifications is the main obstacle hindering the use of cryptographic protocols in VANETs. The aforementioned batch approach significantly alleviates the message verification burden and makes our scheme very practical for securing VANETs. This approach is particularly useful for message verifications when the vehicular density is high.

C. Batch-Message Processing

We next sketch a batch-message-processing procedure by invoking a batch-signature-verification algorithm. Let n distinct messages $\{m'_1, \ldots, m'_n\}$ be newly received for verification during period τ, and let t_i be the threshold for message m'_i. Each message is appended a number of group signatures. Without loss of generality, we assume that *group signatures for each message come from different vehicles* because the group signatures from the same vehicle can be identified due to the message-linkability feature. The batch-message-processing procedure is described as follows.

1) If the TTL of m_i has expired, then the message and its signatures are discarded, and the processing of the message is viewed as *finished*.
2) If the number of group signatures for m_i is greater than t_i and the TTL has not yet expired, then t_i of the signatures will randomly be chosen for batch verification.
   a) If the signatures do not pass the verification, then the message and all of its signatures are discarded.
   b) If the signatures pass the verification, then the message is accepted, and the processing of the message is viewed as *finished*. If necessary, a copy of the message and the t_i group signatures is sent to neighboring vehicles.
3) If the number of group signatures of m_i is smaller than t_i and the TTL has not yet expired, then all the signatures will be chosen for batch verification.
   a) If the signatures do not pass the verification, then the message and all of its signatures are discarded.
   b) If the signatures pass the verification, then the message and the signatures are kept. If necessary, a copy of them is sent to neighboring vehicles.

c) Count the number of group signatures for m_i that passed the verification in the current period and the previous rounds. If the sum is $\geq t_i$, then the message is accepted, and the processing of m_i is viewed as *finished*. Else, wait for more group signatures for the message, and verify them in the next period.

From the aforementioned batch message processing, the verifying vehicle trusts message m_i if it receives at least t_i valid group signatures for m_i before m_i expires. Then, the processing of the message is viewed as finished. Although the verifying vehicle will not trust message m_i if it has not been able to select t_i or more valid group signatures of m_i when m_i expires, the processing of m_i is also finished.

V. Simulation

In this section, we report the results of simulations conducted to evaluate the efficiency, effectiveness, and applicability of the proposed scheme with batch message processing. Another goal of the simulation is to see the relationship between performance and the traffic conditions, the threshold, and the batch-verification period.

A. Simulation Setup

The network simulator ns-2 [39] was used. The VANET scenario was built using the scenario generator presented in [40]. The vehicles were randomly generated, and their average speed was 56 km/h, which is typical in urban areas.[2] The communication range is from 10 to 300 m. The road network considered covers an area of 2.4×2.4 km^2 and is shown in Fig. 2. The channel bandwidth bound is 6 Mb/s, and the package size is 237 B. The TTL of messages is set to 20 s, and the duration of each experiment is 200 s. TTL $= 20$ is an experimental value to test the performance when the requirement to process messages is almost real time. The delay introduced by cryptographic operations is considered in the ns-2 simulation through the measurement of the cryptographic library Multiprecision Integer and Rational Arithmetic C/c++ Library [33]. Experiments were performed on a Pentium 4 PC with a 2-GHz central processing unit, 1-GB random access memory, and a Windows XP operating system.

With the cryptographic library MIRACL [33] in the aforementioned PC environment, the time to generate a group signature is about 0.5 ms, and a group signature verification takes about 5 ms. Hence, in the simulation, we let the batch verification period range from 10 to 60 ms. Note that this period must be long enough to conduct a run of the batch message-processing procedure and the message-generation procedure. Since the speed of cryptographic algorithms is highly dependent on the hardware in use, and there are no related specifications in existing VANET standards, as in [5], we assume a PC-like computational environment, and the simulations concerning performance are merely indicative. However, this assumed environment does not seem to significantly deviate from the onboard computational environment expectable in a vehicle (PCs are affordable and adequate as onboard computers).

[2]Experiments were also performed at different average speeds, varying from 40 to 120 km/h, and we obtained similar results in the experiments. We observe that this is due to the fact that, even for the highest relative speed of 240 km/h, the connection time between two vehicles is at least 4 s. This connection duration is much longer than the time of 0.3 ms needed to transmit a V2V message of 237 B via a channel of 6 Mb/s. This implies that the vehicle speed has little impact on the system performance. Hence, to highlight the main factors affecting the system performance, we only include the simulation results at the typical speed in urban areas.

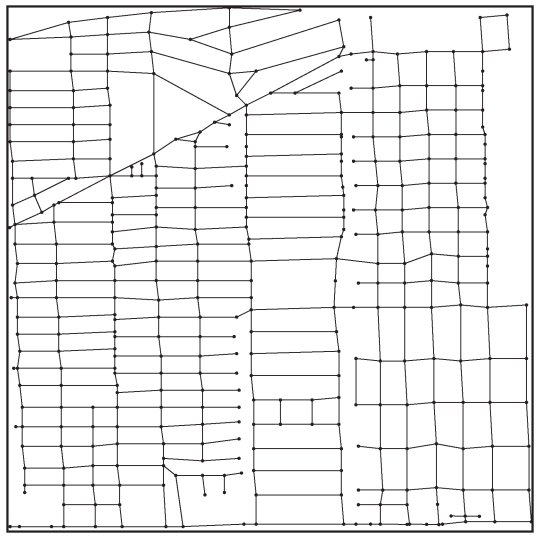

Fig. 2. Road scenario in simulation.

B. Performance Metrics

The performance metrics considered are the average message validation probability P_{msg}, the average message delay D_{msg}, and the average message loss ratio L_{msg}. The message validation probability P_{msg} is the probability that a valid message m_i is endorsed by at least the required threshold t_i of vehicles. The average message delay D_{msg} is the average time latency for a message to be processed (but the processing might be unfinished; see Section IV about batch message processing) after it has been sent from one vehicle to another one within a one-hop communication range. D_{msg} must be smaller than the maximum allowable end-to-end transmission delay [24] because some messages (e.g., related to a serious traffic jam) need to be forwarded to other vehicles. The average message loss ratio L_{msg} is the average possibility that a message cannot be processed before it expires. These metrics measure the effectiveness of the system and depend on the vehicle density (in vehicles per square kilometer), the threshold t, and the batch-verification period τ.

Let v_i be the number of vehicles that receive $\geq t_i$ group signatures on m_i from different vehicles before m_i expires. Let V_i be the total number of vehicles processing m_i (those who have received m_i for validation). For n messages, the average message validation probability P_{msg} is $P_{\text{msg}} = (1/n)\sum_{i=1}^{n}(v_i/V_i)$. The aforementioned probability is computed for different thresholds and vehicle densities.

The average message delay D_{msg} is defined as follows: $D_{\text{msg}} = (1/L_{\mathbb{D}})\sum_{\ell\in\mathbb{D}}((1/M_{\ell_\rightarrow})\sum_{m=1}^{M_{\ell_\rightarrow}}(T_{\text{sgn}}^{\ell_m} + (1/K_\ell)\sum_{k=1}^{K_\ell}(T_{\text{trnsmsn}}^{\ell_m k} + \sum_{j=1}^{\lceil MAD/\tau\rceil} jP_{m,j}\tau)))$, where $\mathbb{D}$ is

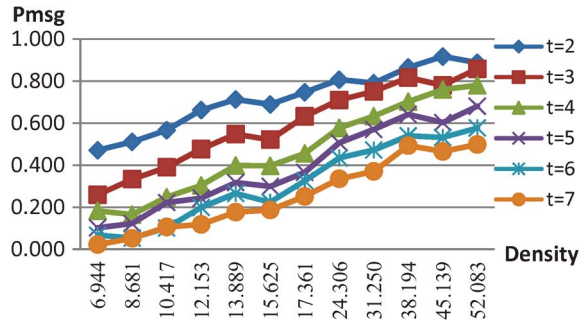

Fig. 3. Impact of authentication on the message-verification rate.

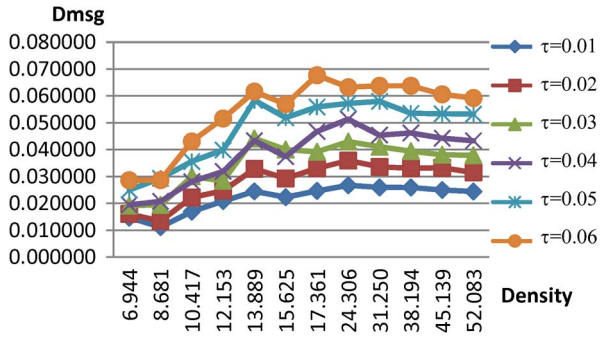

Fig. 4. Impact of authentication on the message delay.

the sample district (area) in the simulation, $L_{\mathbb{D}}$ is the number of vehicles in $\mathbb{D}$, $M_{\ell\rightarrow}$ is the number of messages sent by vehicle ℓ, K_ℓ is the number of vehicles within a one-hop communication range of vehicle ℓ, $T_{\overrightarrow{\text{sgn}}}^{\ell_m}$ is the time taken by vehicle ℓ for signing message m, $T_{\overrightarrow{\text{trnsmsn}}}^{\ell_m k}$ is the time taken to transmit message m from vehicle ℓ to vehicle k, τ is the period taken to do a batch verification, and MAD is the maximum allowable delay for end-to-end message transmission. According to [24], MAD is 100 ms. $P_{m,j} = (v_{m,j}/V_m)$, where V_m is the total number of vehicles processing m among the K_ℓ vehicles, and $v_{m,j}$ is the number of vehicles processing m in the interval $((j-1)\tau, j\tau]$ for $j\tau \leq MAD$. Clearly, we have that $V_m = \sum v_{m,j}$.

The average message loss ratio reflects the applicability of the scheme. The average message loss ratio L_{msg} is defined by $L_{\text{msg}} = 1 - (1/L_{\mathbb{D}}) \sum_{\ell=1}^{L_{\mathbb{D}}} (M_{\text{consumed}}^{\ell} / \sum_{k=1}^{K_\ell} M_{k\rightarrow\ell})$, where $M_{\text{consumed}}^{\ell}$ is the number of messages consumed by vehicle ℓ in the application layer, and $M_{k\rightarrow\ell}$ is the number of messages that have been sent to vehicle ℓ in the medium-access control layer. A message will be lost if it cannot reach its destination or the queue is full (e.g., due to a message arrival rate higher than the message processing rate). When computing the message-loss ratio, we should notice that, if the t signatures of message m have been verified, the processing of all copies of m is viewed as finished [even the newly arriving copies of m (see Section IV about batch message processing)]. The message loss ratio is computed for different verification periods and vehicle densities.

C. Simulation Analysis

Fig. 3 shows the average probability $\mathtt{P_{msg}}$ of a message being validated by vehicles for a fixed TTL of 20 s, several values of threshold t, and several values of vehicle density d expressed in vehicles per square kilometer. This verification probability indicates the availability offered by the protocol. A higher verification probability implies that most vehicle-generated messages can get verified and then be used to guide other vehicles in making decisions. The threshold t indicates the trustworthiness of messages. A higher threshold means that more vehicles observe the same situation and agree with the message; hence, more confidence is gained in the trustworthiness of the message after validating it with the verification procedure. We also observe that the message verification probability may grow with the TTL of a message because a longer TTL implies that vehicles in the vicinity have more time to wait for enough endorsed copies of the message. However, since the TTL is determined by message types and cannot be adjusted, for simplicity, we used a fixed and short (near-real-time) TTL $= 20$ in our experiments.

From Fig. 3, the results of the experiment show that the average verification probability of messages depends on the threshold and the vehicle density. For a given threshold, the verification probability almost linearly grows with the vehicle densities. We observe that this is natural because, with more vehicles in the area, a vehicle may receive more endorsed copies of the same message up to the required threshold. If the vehicle density is fixed, then the verification probability degrades as the threshold grows. This is reasonable because, if the threshold is high, it is possible that some vehicles cannot receive enough endorsed copies before the message expires. The experiment shows that, by adjusting the threshold for different vehicle densities, we can achieve a stable message verification probability, which should yield good availability of the proposed protocol for different traffic conditions. This feature allows adaptively changeable threshold for message authentications in VANETs.

Another important factor that determines the performance of the security protocol is the latency introduced by verifying the received copies of the message endorsed by other vehicles. The latency is measured by average message delay $\mathtt{D_{msg}}$ and computed for different batch-verification periods τ and traffic conditions but fixed threshold $t = 5$ and message lifetime $TTL = 20$ s. Clearly, a lower latency implies that it takes vehicles less time to respond to the reported changes in the traffic environment. As a result, the corresponding VANET improves traffic efficiency and reduces traffic accidents.

The simulation results are given in Fig. 4. It can be seen that, given a fixed batch-verification period, the average message delay increases as the vehicle density grows and reaches the maximum value when the vehicle density is between 17 and 24 vehicles/km^2. After that, the average message delay no longer grows, although vehicles received increasing messages to be verified. This feature implies that our protocol is applicable for various traffic conditions and that its performance does not seriously degrade in the case of a high density of vehicles. We observe that this feature is due to the proposed batch-message-processing method, which allows verifying a batch of messages as a single one. Furthermore, with the increase in the vehicle density, it takes less time for vehicles to wait for copies of a message up to the threshold.

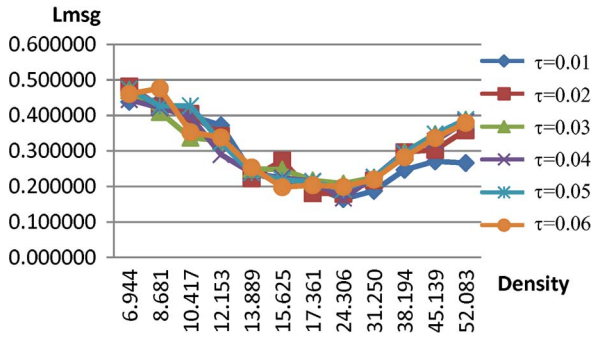

Fig. 5. Impact of authentication on the message loss rate.

The average message delay is also affected by the batch-verification period. As illustrated in Fig. 4, for a fixed vehicle density, the average message delay increases as the batch-verification period increases. This is rational because vehicles can only perform the verification procedure after the last verification period is over. A longer verification period implies that each vehicle spends more time on waiting for the next verification period. Hence, the shorter verification period yields less average message delay. However, the shortest verification period has to be at least as long as the time for a vehicle to perform a batch verification.

Finally, we evaluate the average message loss ratio L_{msg} of the proposed protocol. The simulation results are given in Fig. 5. The average message loss ratio indicates the applicability of the protocol. If the loss ratio is too high, then most reported messages cannot be exploited by other vehicles to improve traffic efficiency and safety, even though these messages may be true and endorsed by many vehicles.

From Fig. 5, it is notable that the average message-loss ratio does not always increase as the vehicle density increases. In the normal vehicle density of 14 to 32 vehicles/km^2, the message-loss ratio is about 20%. If the vehicles are too sparse or dense, then the message-loss ratio can be as high as 50%. However, if the vehicles are very sparse, then vehicles may have better driving conditions (e.g., few vehicles in the road), and the loss of messages will not seriously affect the driving efficiency or safety, just like in a good driving environment without a VANET. For a heavy traffic load, it is also acceptable if a large number of messages are lost because most of the messages are repeatedly sent by vehicles.

It can also be seen that the average message-loss ratio does not vary a lot with the decrease in the batch-verification period. This demonstrates robustness of the proposed protocol to different batch verification strategies. This stability allows us to optimize the performance of our authentication protocol by setting the batch-verification period as short as possible without degrading the applicability of the protocol.

VI. Conclusion

In this paper, we have proposed a new efficient system for balancing public safety and vehicle privacy in VANETs. Both *a priori* and *a posteriori* countermeasures have been used to thwart attackers. We have achieved this goal by drawing on the novel technology of MLGSs. We have realized a context-aware threshold-authentication scheme for V2V communications in which the threshold can adaptively change in light of the context of messages, rather than having to be preset during the system-design stage. Furthermore, a fast batch-verification method has been presented to speed up the validation of authenticated messages. Such a batch-verification approach is critical to make authentications implementable in VANETs, since vehicles in those networks periodically receive a large number of messages to be validated. Extensive simulation has been conducted to evaluate the average message-verification probability, latency, and message-loss ratio of our protocol. The experiments demonstrate that our scheme can achieve s stable message-verification probability by setting different thresholds for different traffic conditions. The experiments also illustrate that our scheme can achieve a quite low latency and message-loss ratio. These features leads to the applicability of our scheme for various V2V communication authentications while preserving privacy for honest vehicles.

Appendix A
Proof of Claim 1

Proof: We show that any message generated by following the protocol will be accepted by the verification procedure. From the vehicle-registration procedure in Section III-E, we have that $K_1 = g_1^k$ and $K_2 = Z(h_1 Y)^{-k}$. From the message generation procedure in Section III-F, we have that $\sigma_1 = K_1 g_1^s$, $\sigma_2 = K_2 (h_1 Y)^{-s}$, and $\sigma_3 = \sigma_1^y$. If we set $k' = k + s$, it follows that $\sigma_1 = g_1^{k+s} = g_1^{k'}$ and $\sigma_2 = Z(h_1 Y)^{-(k+s)} = Z(h_1 Y)^{-k'}$. Since g_1 is a generator of $\mathbb{G}_1$, there exist some unknown values α, $\beta \in \mathbb{Z}_p^*$ such that $h_1 = g_1^\alpha$ and $U_1 = g_1^\beta$. Note that ϕ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, $\phi(g_2) = g_1$, $\phi(h_2) = h_1$, and $\phi(U_2) = U_1$. Therefore, it follows that $h_2 = g_2^\alpha$ and that $U_2 = g_2^\beta$. Thus, we have that $e(\sigma_2, g_2) e(\sigma_1, h_2) e(\sigma_3, U_2) = Ae(g_1^{-(\alpha+y\beta)k'}, g_2) e(g_1^{(\alpha+y\beta)k'}, g_2) = A$. Hence, the first equation in Section III-G holds.

From Section III-F, we have that $R_1 = H_1(m)^r$, $R_2 = \sigma_1^r$, $\sigma_4 = H_1(m)^y$, and $\sigma_6 = r - \sigma_5 y \mod p$. It follows that

$$R_1 = H_1(m)^{\sigma_6 + \sigma_5 y} = H_1(m)^{\sigma_6} \sigma_4^{\sigma_5} \tag{9}$$

$$R_2 = \sigma_1^{\sigma_6 + \sigma_5 y} = \sigma_1^{\sigma_6} \left(H_1(m)^y\right)^{\sigma_5} = \sigma_1^{\sigma_6} \sigma_3^{\sigma_5}. \tag{10}$$

Furthermore, according to Section III-F, we have

$$\sigma_5 = H\left(m \| \sigma_1 \| \sigma_2 \| \sigma_3 \| \sigma_4 \| R_1 \| R_2\right). \tag{11}$$

By replacing R_1 and R_2 in (11) with (9) and (10), respectively, we further have that

$$\sigma_5 = H\left(m \| \sigma_1 \| \sigma_2 \| \sigma_3 \| \sigma_4 \| H_1(m)^{\sigma_6} \sigma_4^{\sigma_5} \| \sigma_1^{\sigma_6} \sigma_3^{\sigma_5}\right).$$

Hence, the second equation in Section III-G holds.

So far, we have shown that both verifications hold if the message has been generated by following the protocol. Hence, the scheme is correct. ■

APPENDIX B
PROOF OF CLAIM 2

Proof: We will show that if an attacker could efficiently forge a valid group signature in our proposal, then the DHK assumption in $\mathbb{G}_1$ of pairing groups Υ would be broken by invoking the attacker as a black box. Hence, if the DHK assumption holds in pairing groups, then such an attacker forging valid vehicle messages cannot exist, and the aforementioned authentication scheme is unforgeable. We next go into the details.

Assume that there exists an attacker $\mathcal{A}$ who controls a number of legal vehicles and can forge a message that is accepted by other vehicles without the controlled vehicles being traceable, given that the attacker can request the public system parameters of the scheme and a polynomial number of valid vehicle-generated messages.

To answer the queries from the attacker, we run the DHK challenger in pairing groups $\Upsilon = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e) \leftarrow \mathtt{PGen}(1^\lambda)$, where the DDH and DHK assumptions hold (see Section III-A), and let ϕ be a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ such that $\phi(g_2) = g_1$. Then, we obtain a DHK challenge $(g_2, h_2) = (g_2, g_2^\alpha) \in \mathbb{G}_2^2$, where $\alpha \in \mathbb{Z}_p^*$ is unknown to us. We compute (g_1, h_1), where $h_1 = \phi(h_2)$. Note that we do not know the value $\alpha \in \mathbb{Z}_p^*$ such that $h_1 = g^\alpha$. Let h_2 and U_2 be randomly chosen from $\mathbb{G}_2$. Hence, $\phi(h_2) = h_1$ and $\phi(U_2) = U_1$ can efficiently be computed. We randomly choose two hash fuctions $H_1(\cdot) : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H(\cdot) : \{0,1\}^* \rightarrow \mathbb{Z}_p$, which will be modeled as random oracles. That is, the outputs of $H(\cdot)$ and $H_1(\cdot)$ are set as random values, and the attacker has to query the oracles of both functions to compute their outputs. The simulation of hash functions is standard for signatures as in [41]. Then, we obtain system parameters $\pi = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e; h_2, h_1; U_2, U_1; H_1, H \rangle$.

When the attacker $\mathcal{A}$ requires the system parameters of the scheme, we forward π to the attacker. When the attacker requires $\mathcal{RM}$'s public key A, we generate A as in the real scheme and send it to $\mathcal{A}$. When the attacker requests a group certificate with Y and the signature of Y from $\mathcal{TM}$, we run a zero-knowledge proof $ZK\{y|Y = g_1^y\}$ with the attacker. From the forking lemma in [41], we can efficiently extract y by invoking $\mathcal{A}$ twice. Since we know $\mathcal{RM}$'s secret key Z to satisfy $e(Z, g_2) = A$ and we know the secret keys of the registered public keys, we can generate group certificates and answer the attacker with valid messages as in the real scheme.

At some point, the attacker $\mathcal{A}$ forges a valid vehicle-generated message that contains a group signature $\sigma = (\sigma_1, \ldots, \sigma_6)$ on the first five fields of the message, where $e(\sigma_2, g_2) = e(\sigma_1, T)$ does not hold for any registered T. Since the forged message is valid, the verification equations $e(\sigma_2, g_2) = Ae(\sigma_1, h_2)e(\sigma_3, g_2)$ and $\sigma_5 = H(m\|\sigma_1\|\sigma_2\|\sigma_3\|\sigma_4\|H_1(m)^{\sigma_6}\sigma_4^{\sigma_5}\|\sigma_1^{\sigma_6}\sigma_3^{\sigma_5})$ hold. Note that the second equation implies that $(\sigma_1, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$ is a signature on message m under a one-time public key $(\sigma_3 = \sigma_1^{y'}, \sigma_4 = H_1(m)^{y'})$. Again, from the forking lemma in [41], we can efficiently extract y' by invoking $\mathcal{A}$ twice. Hence, we obtain that $e(\sigma_2, g_2)e(\sigma_1, h_2)e(\sigma_1^{y'}, U_2) = A$ and $e(\sigma_2, g_2) = Ae(\sigma_1^{-1}, h_2)e(\sigma_1^{-y'}, U_2)$.

Since g_1 and h_1 are generators of $\mathbb{G}_1$, there exists $\alpha, k' \in \mathbb{Z}_p^*$ satisfying $\sigma_1 = g_1^{k'}$ and $h_1 = g_1^\alpha$, where α and k' are unknown to us since h_1 comes from the DHK challenger, and $\mathcal{A}$ is a black box. Note that $\phi(h_2) = h_1$ and $\phi(g_2) = g_1\phi(U_2) = U_1$ for the isomorphism ϕ. We have $h_2 = g_2^\alpha$ and $U_2 = g_2^u$. It follows that $e(\sigma_2, g_2) = e(Z, g_2)e(g_1^{-k'}, g_2^\alpha)e((g_1^{k'})^{-y'}, g_2^u) = e(Z, g_2)e((g_1^\alpha)^{-k'}, g_2)e((g_1^{uy'})^{-k'}, g_2) = e(Zh_1^{-k'}\sigma_1^{-uy'}, g_2)$. Hence, we have that $\sigma_1 = g_1^{k'}$ and that $\sigma_2 = Zh_1^{-k'}\sigma_1^{-uy'}$.

Since we can extract y' by running $\mathcal{A}$, we obtain $\sigma_2' = \sigma_2\sigma_1^{uy'} = Zh_1^{-k'}$, where k' is unknown to us. Note that (σ_1, σ_2') is an El Gamal ciphertext of Z under the public key h_1. Hence, $(g_1, h_1, \sigma_1, Z/\sigma_2') = (g_1, h_1, g_1^{k'}, h_1^{k'})$ is a Diffie–Hellman tuple, where k' is unknown to us. This implies that $(g_2, h_2, \sigma_1, Z/\sigma_2')$ is an accompanied Diffie–Hellman tuple, and we can use it to answer the DHK challenge. This contradicts the DHK assumption in $\mathbb{G}_1$. ■

APPENDIX C
PROOF OF CLAIM 3

Proof: We prove that if there is an attacker $\mathcal{A}$ that can successfully break the anonymity of the scheme, given that the attacker can request the public system parameters of the scheme, $\mathcal{RM}$'s public key, $\mathcal{TM}$'s public key, the vehicle public keys, and group signatures under the public keys of registered vehicles, then we can convert this attacker into an efficient algorithm to break the DDH assumption in $\mathbb{G}_1$. This is impossible if the DDH assumption holds.

We are given a DDH challenge $(g, h, g^y, h^z) = (g_1, h_1, g_1^y, h_1^z)$ in $\mathbb{G}_1$ of pairing groups $\Upsilon = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e) \leftarrow \mathtt{PGen}(1^\lambda)$ explained in Section III-A, and we let ϕ be a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ such that $\phi(g_2) = g_1$. Let h_2 and U_2 be randomly chosen from $\mathbb{G}_2$. Hence, $\phi(h_2) = h_1$ and $\phi(U_2) = U_1$ can efficiently be computed. We randomly choose two hash functions $H_1(\cdot) : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H(\cdot) : \{0,1\}^* \rightarrow \mathbb{Z}_p$, which will be modeled as random oracles. Then, we obtain system parameters $\pi = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e; h_2, h_1; U_2, U_1; H_1, H \rangle$.

We are also required to answer the attacker's queries for the system parameters, $\mathcal{RM}$'s public key, $\mathcal{TM}$'s public key, vehicle public keys, and group signatures under the public keys of registered vehicles. We will give the attacker one challenge group signature. The attacker has to answer which vehicle's public key has been used to generate the group signature. Finally, we will use the attacker's guess bit to answer whether $z = y$. We next go into the details.

When the attacker $\mathcal{A}$ requires the system parameters of the scheme, we forward π to the attacker. When the attacker requires $\mathcal{RM}$'s public key A, we generate it as in the real scheme and send it to $\mathcal{A}$. When the attacker requires $\mathcal{TM}$'s public key, we randomly choose an element in that public key space and send it to the attacker. Finally, we randomly select $b \in \{0,1\}$, $y' \in \mathbb{Z}_p^*$ and set $Y_b = (g^y)^u$ and $Y_{b\oplus 1} = U_1^{y'}$ as the public keys of the two vehicles. Note that we do not know y.

The hash functions are modeled as random oracle models. Without loss of generality, we assume that the attacker queries H_1 with different messages at most q_{H_1} times. To simulate H_1,

we randomly choose $1 \leq i^* \leq q_{H_1}$. For the ith $(i \neq i^*)$ query $m_i \in \{0,1\}^*$, we randomly select $\alpha_{m_i} \in \mathbb{Z}_p^*$ and set $h_1(m_i) = g^{\alpha_{m_i}}$. For the i^*th query $m_{i^*} \in \{0,1\}^*$, we randomly select $\alpha_{m_{i^*}} \in \mathbb{Z}_p^*$ and set $H_1(m_{i^*}) = h^{\alpha_{m_{i^*}}}$. We maintain a table to record $(i, m_i, \alpha_{m_i}, H_1(m_i))$ for consistency.

When the attacker requests a group signature on m that can be traced to Y_b, we declare a failure if $m = m_{i^*}$. This bad event happens with a probability of $1/q_{H_1}$. If the attacker requests a group signature on m, which can be traced to Y_b but $m \neq m_{i^*}$, we randomly select $a \in \mathbb{Z}_p^*$ and compute $\sigma_1 = g_1^a$, $\sigma_2 = Z(h_1 Y_b)^{-a}$, $\sigma_3 = Y_0^a = \sigma_1^a$, and $\sigma_4 = Y_b^{\alpha_m/u} = H_1(m)^y$.

We compute σ_5 and σ_6 with the standard simulation technique for signatures [41]. This simulation does not need knowledge of y (which we do not have). If the attacker requests a group signature on m that can be traced to $Y_{b\oplus 1}$, we can compute it as in the real scheme since we know the corresponding secret keys. Hence, if $m \neq m_{i^*}$, we can always answer the attacker with valid group signatures.

At some point, the attacker requests a challenge group signature on a message m^*. If $m^* \neq m_{i^*}$, we declare a failure. This bad event happens with probability $1 - (1/q_{H_1})$. Else, we produce the challenge signature by computing $\sigma_1^* = h$, $\sigma_2^* = Zh^{-\delta}(h^z)^{-1}$, $\sigma_3^* = h^z$, and $\sigma_4^* = \sigma_3^{\alpha_{m^*}} = H_1(m^*)^z$. Again, we compute σ_5^* and σ_6^* with a standard simulation technique [41] of signatures. It is straightforward to verify that the challenge signature satisfies the two verification equations in Section III-G and is a valid group signature. Let the attacker's guess bit be b'. Note that the challenge group signature will be traced to Y_b if and only if $y = z$. Hence we claim that $y = z$ if and only if $b' = b$. If the attacker has a success probability of $(1/2) + \varepsilon$, then we have a success probability of $(1/2) + (1/q_{H_1})(1 - (1/q_{H_1}))\varepsilon$. If ε is non-negligible, then $(1/q_{H_1})(1 - (1/q_{H_1}))\varepsilon$ is non-negligible (since q_{H_1} is a polynomial in λ), which contradicts the DDH assumption in $\mathbb{G}_1$. Hence, ε is negligible, and this completes the proof. ■

ACKNOWLEDGMENT

The authors are with the United Nations Educational, Scientific, and Cultural Organization (UNESCO) Chair in Data Privacy, but their views do not necessarily reflect the position of UNESCO nor commit that organization.

REFERENCES

[1] J. Blau, "Car talk," *IEEE Spectr.*, vol. 45, no. 10, p. 16, Oct. 2008.
[2] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ASTM E2213-03, Sep. 2003.
[3] F. Dötzer, "Privacy issues in vehicular ad hoc networks," in *Proc. PET*, vol. 3856, *Lecture Notes in Computer Science*, 2005, pp. 197–209.
[4] J. Guo, J. P. Baugh, and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," in *Proc. Mobile Netw. Veh. Environ.*, 2007, pp. 103–108.
[5] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.
[6] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *Wireless Commun.*, vol. 13, no. 5, pp. 8–15, Oct. 2006.
[7] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng, "Cross-layer privacy enhancement and non-repudiation in vehicular communication," in *Proc. 4th WMAN*, Bern, Switzerland, Mar. 2007.
[8] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.—Special Issue Security Ad Hoc Sensor Networks*, vol. 15, no. 1, pp. 39–68, 2007.
[9] M. E. Zarki, S. Mehrotra, G. Tsudik, and N. Venkatasubramanian, "Security issues in a future vehicular network," in *Proc. Eur. Wireless*, 2002, pp. 270–274.
[10] M. Raya and J. Hubaux, "The security of vehicular ad hoc networks," in *Proc. 3rd ACM Workshop Security Ad Hoc Sensor Netw.*, Alexandria, VA, Nov. 2005, pp. 11–21.
[11] M. Raya, A. Aziz, and J.-P. Hubaux, "Efficient secure aggregation in VANETs," in *Proc. 3rd VANET*, 2006, pp. 67–75.
[12] F. Picconi, N. Ravi, M. Gruteser, and L. Iftode, "Probabilistic validation of aggregated data in vehicular ad-hoc networks," in *Proc. 3rd VANET*, 2006, pp. 76–85.
[13] IEEE P1609.2 Version 1—Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages, 2006.
[14] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1557–1568, Oct. 2007.
[15] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "CARAVAN: Providing location privacy for VANET," in *Proc. ESCAR*, Nov. 2005.
[16] G. Calandriello, P. Papadimitratos, A. Lioy, and J.-P. Hubaux, "Efficient and robust pseudonymous authentication in VANETs," in *Proc. VANET*, 2007, pp. 19–28.
[17] E. Eiland and L. Liebrock, "An application of information theory to intrusion detection," in *Proc. IWIA*, 2006, pp. 119–134.
[18] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to DDoS attack detection and response," in *Proc. DARPA Inf. Survivability Conf. Expo.*, 2003, pp. 303–314.
[19] W. Lee and D. Xiang, "Information-theoretic measures for anomaly detection," in *Proc. IEEE Symp. Security Privacy*, 2001, pp. 130–143.
[20] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proc. 1st ACM Int. Workshop Veh. Ad Hoc Netw.*, 2004, pp. 29–37.
[21] B. Ostermaier, F. Dötzer, and M. Strassberger, "Enhancing the security of local danger warnings in VANETs—A simulative analysis of voting schemes," in *Proc. 2nd Int. Conf. Availability, Rel. Security*, 2007, pp. 422–431.
[22] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proc. ACM Workshop Hot Topics Netw.*, 2005.
[23] V. Daza, J. Domingo-Ferrer, F. Sebe, and A. Viejo, "Trustworthy privacy-preserving car-generated announcements in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1876–1886, May 2009.
[24] 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.
[25] J. Domingo-Ferrer and Q. Wu, "Safety and privacy in vehicular communications," in *Proc. Privacy Location-Based Appl.*, vol. 5599, *Lecture Notes in Computer Science*, Berlin, Germany, 2009, pp. 173–189.
[26] G. Kounga, T. Walter, and S. Lachmund, "Proving reliability of anonymous information in VANETs," *IEEE Trans. Veh. Technol.*, vol. 58, no. 6, pp. 2977–2989, Jul. 2009.
[27] D. Chaum and E. van Heyst, "Group signatures," in *Proc. Eurocrypt*, vol. 547, *Lecture Notes in Computer Science*, New York, 1991, pp. 257–265.
[28] T. Nakanishi, T. Fujiwara, and H. Watanabe, "A linkable group signature and its application to secret voting," *Trans. Inf. Process. Soc. Jpn.*, vol. 40, no. 7, pp. 3085–3096, 1999.
[29] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement," in *Proc. Eurocrypt*, vol. 5479, *Lecture Notes in Computer Science*, New York, 2009, pp. 153–170.
[30] S. D. Galbraith, K. G. Paterson, and N. P. Smart, Pairings for cryptographers. [Online]. Available: http://eprint.iacr.org/2006/165.pdf
[31] I. B. Damgård, "Towards practical public key systems secure against chosen ciphertext attacks," in *Proc. Crypto*, vol. 576, *Lecture Notes in Computer Science*, New York, 1991, pp. 445–456.
[32] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Proc. Crypto*, vol. 740, *Lecture Notes in Computer Science*, 1992, pp. 89–105.
[33] Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL). [Online]. Available: http://www.shamus.ie/
[34] X. Boyen and B. Waters, "Compact group signatures without random oracles," in *Proc. Eurocrypt*, vol. 4004, *Lecture Notes in Computer Science*, 2006, pp. 427–444.

[35] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen, "Batch verification of short signatures," in *Proc. Eurocrypt*, vol. 4515, *Lecture Notes in Computer Science*, 2007, pp. 246–263.
[36] J. Pastuszak, D. Michatek, J. Pieprzyk, and J. Seberry, "Identification of bad signatures in batches," in *Proc. PKC*, vol. 3958, *Lecture Notes in Computer Science*, 2000, pp. 28–45.
[37] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: A robust signature scheme for vehicular communications using Binary Authentication Tree," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1974–1983, Apr. 2009.
[38] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen, On the practicality of short signature batch verification. [Online]. Available: http://eprint.iacr.org/2008/015.pdf
[39] The Network Simulator—ns. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/Main Page
[40] A. K. Saha and D. B. Johnson, "Modeling mobility for vehicular ad hoc networks," in *Proc. 1st VANET*, 2004, pp. 91–92.
[41] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, Dec. 2000.

Qianhong Wu received the M.Sc. degree in applied mathematics from Sichuan University, Sichuan, China, in 2001 and the Ph.D. degrees in cryptography from Xidian University, Xian, China, in 2004.

He was an Associate Research Fellow with Wollongong University, Wollongong, Australia. He is currently an Associate Professor with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Computers, Wuhan University, Wuhan, China. He is also a Senior Researcher with the United Nations Educational, Scientific, and Cultural Organization Chair in Data Privacy, Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Tarragona, Spain. He is the author of more than 60 publications. He has been a Main Researcher or Project holder/Coholder of more than ten Chinese-, Australian-, and Spanish-funded projects. His research interests include cryptography, information security and privacy, and ad hoc network security.

Dr. Wu is a member of the International Association for Cryptologic Research. He has served on the program committees of several international conferences on information security and privacy.

Josep Domingo-Ferrer (SM'02) received the M.Sc. and Ph.D. degrees (with honors—Outstanding Graduation Award) in computer science from the Universitat Autnoma de Barcelona, Barcelona, Spain, in 1988 and 1991, respectively, and the M.Sc. degree in mathematics from the Universidad Nacional de Educación a Distancia, Madrid, Spain.

He is a Full Professor of computer science and an Catalan Institution for Research and Advanced Studies (ICREA) Acadèmia Researcher with the Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Tarragona, Spain, where he holds the United Nations Educational, Scientific, and Cultural Organization (UNESCO) Chair in Data Privacy. In 2004, he was a Visiting Fellow with Princeton University, Princeton, NJ. He is the holder of three patents. He is the author of more than 225 publications, one of which became an Institute for Scientific Information highly cited paper in early 2005. He has been the Coordinator of European Union FP5 project CO-ORTHOGONAL and several Spanish-funded and U.S.-funded research projects. He currently coordinates the CONSOLIDER "ARES" team on security and privacy: one of Spain's 34 strongest research teams. His fields of activity are data privacy, data security, and cryptographic protocols.

He has received three research awards and four entrepreneurship awards, among them the ICREA Acadèmia Research Prize from the Government of Catalonia. He has chaired or cochaired nine international conferences and has served on the program committees of more than 70 conferences on privacy and security. He is a Co-Editor-in-Chief of the IEEE TRANSACTIONS ON DATA PRIVACY and an Associate Editor of three international journals.

Úrsula González-Nicolás received the M.Sc. degree in computer engineering from Universitat Rovira i Virgili, Tarragona, Spain, in 2009. She is currently working toward the Master's degree in information and security engineering from the United Nations Educational, Scientific, and Cultural Organization Chair in Data Privacy, Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili.

Her graduation project was focused on genetic algorithms. Her research interest is information privacy.