DCS: An Efficient Distributed-Certificate-Service Scheme for Vehicular Networks

Albert Wasef, *Member, IEEE*, Yixin Jiang, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—In this paper, we propose an efficient distributed-certificate-service (DCS) scheme for vehicular networks. The proposed scheme offers flexible interoperability for certificate service in heterogeneous administrative authorities and an efficient way for any onboard units (OBUs) to update its certificate from the available infrastructure roadside units (RSUs) in a timely manner. In addition, the DCS scheme introduces an aggregate batch-verification technique for authenticating certificate-based signatures, which significantly decreases the verification overhead. Security analysis and performance evaluation demonstrate that the DCS scheme can reduce the complexity of certificate management and achieve excellent security and efficiency for vehicular communications.**

***Index Terms*—Batch verification, certificate service, communication security, revocation, vehicular networks.**

I. Introduction

RECENTLY, vehicular ad hoc networks (VANETs) have attracted extensive attention for their promise in revolutionizing transportation systems. VANETs consist of network entities, mainly including vehicles and roadside units (RSUs). Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications are two basic vehicular communication modes, which, respectively, allow vehicles to communicate with each other or with the roadside infrastructure.

Due to the open-medium nature of wireless communications and the high-speed mobility of a large number of vehicles in spontaneous vehicular communications, entity authentication, message integrity, nonrepudiation, and privacy preservation are identified as primary security requirements [1], [2]. It is evident that any malicious behavior of a user, such as injecting false information and modifying and replaying the disseminated messages, could be fatal to other legal users. Furthermore, the privacy of users must be guaranteed in the sense that the privacy-related information of a vehicle should be protected to prevent an observer from revealing the real identities of the users, tracking their locations, and inferring sensitive data [3], [4]. Hence, to satisfy the security and privacy requirements, it is a prerequisite to elaborately design a suite of protocols to achieve security and privacy for practical vehicular networks. A well-recognized solution is to deploy a public-key infrastructure (PKI) [5], where each OBU has a set of authentic certificates. To protect the privacy of users, each OBU should use a certificate for a short duration, and after that, it has to replace this certificate, i.e., OBUs continuously consume their certificate sets. Eventually, each OBU will need to update its certificates. In the classical PKI, any certificate update must be performed through a central certification authority (CA), which sends the updated certificate to the requesting OBU through the available RSUs on the roads. The centralized certificate update process in the classical PKI may be impractical in large-scale VANETs due to the following reasons: 1) Each CA encounters a large number of certificate update requests, which can render the CA with a bottleneck, and 2) the certificate-update delay is long relative to the short V2I communication duration between the immobile RSUs and the highly mobile OBUs, during which the new certificate should be delivered to the requesting OBU. The long certificate-update delay is due to the fact that a request submitted by an OBU to an RSU must be forwarded to the CA, and the CA has to send the new certificate to that RSU, which, in turn, forwards the new certificate to the requesting OBU. Accordingly, the classical PKI should be pruned or optimized to satisfy the certificate-service requirement in volatile vehicular-communication scenarios. To provide a practical certification service for VANETs, it is required for each OBU to efficiently update its certificate in a timely manner. The certification service should also be decentralized to enable VANETs to efficiently process the expected large number of certificate-update requests. Moreover, to protect the user privacy, the updated certificates should be anonymous and free from the key escrow issue.

Another important issue is the roaming between different domains [6], [7]. The OBUs should have the capabilities to roam between domains administered by different CAs. The wireless access in vehicular environments (WAVE) standard [8] does not consider the roaming issue, and the interopreability between different CAs is still an open issue that has not been previously tackled in the VANET literature.

According to the dedicated short-range communication (DSRC) [9], which is part of the WAVE standard, each OBU periodically broadcasts a message every 300 ms, where entity authentication and message integrity can be achieved by verifying the certificate and digital signature of the sender. In dense-traffic areas, each OBU will receive a large number

Manuscript received December 24, 2008; revised June 6, 2009. First published July 31, 2009; current version published February 19, 2010. The review of this paper was coordinated by Dr. J. Deng.

A. Wasef and X. Shen are with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: awasef@bbcr.uwaterloo.ca; xshen@bbcr.uwaterloo.ca).

Y. Jiang is with the Department of Computer Science and Technology, Tsinghua University, Beijing 10084, China (e-mail: yixin.tsinghua@gmail.com).

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.2009.2028893

of messages in a short duration, and thus, the ability to verify a large number of certificates and signatures in a specific period poses an inevitable challenge to the authentication technique.

To address the aforesaid security and performance issues, we introduce an efficient distributed-certificate-service (DCS) scheme for vehicular communications, which features the following properties.

1) *Scalability*: The DCS scheme is constructed in a hierarchical way, which enables any OBU to efficiently update its certificate from the available RSUs in a timely manner. Thus, the DCS scheme offers a distributed certification service and flexible interoperability between different administrative authorities, and it enables the certificates of the OBUs to be free from the key escrow. All such policies efficiently enhance the system scalability, particularly when it is deployed in large-scale and heterogeneous vehicular networks.
2) *Efficiency*: Considering the requirement for each entity to verify a large number of messages in a timely manner, DCS introduces an efficient batch verification technique, which enables any entity to simultaneously verify a mass of signatures and certificates. Thus, the DCS scheme significantly decreases the verification overhead.

Therefore, the DCS scheme can meet the security and efficiency requirements for certificate service in vehicular communications.

The remainder of this paper is organized as follows. In Section II, related works are surveyed. In Section III, the preliminaries are discussed. The system design considerations in the proposed DCS scheme are investigated in Section IV. The proposed DCS scheme is introduced in Section V. Section VI introduces an efficient batch-verification technique for authenticating certificate-based message signatures. Sections VII and VIII present the security analysis and performance evaluation for the proposed DCS scheme, respectively, followed by the conclusion in Section IX.

II. Related Works

In spontaneous vehicular communications, the primary security requirements are identified as entity authentication, message integrity, nonrepudiation, and privacy preservation. Deploying an efficient PKI is a well-recognized solution for achieving security and privacy for practical vehicular networks [1], [5]. Although VANETs have recently gained extensive attention, very few works have addressed the design of a PKI that is suitable for the security requirements of VANETs.

In [5], Hubaux identifies the specific issues of security and privacy challenges in VANETs and claims that a PKI should be well deployed to protect the transited messages and to mutually authenticate among network entities. In [1], Raya and Hubaux use a classical PKI to provide secure and privacy-preserving communications to VANETs. For this approach, each vehicle needs to preload a huge pool of anonymous certificates. The number of the loaded certificates in each vehicle should be large enough to provide security and privacy preservation for a long time, e.g., one year. Each vehicle can update its certificates from a central authority during the annual inspection of the vehicle. The requirement to load a large number of certificates in each vehicle incurs inefficiency for certificate management, as revoking one vehicle implies revoking the huge number of certificates loaded in it.

Lin *et al.* [10] use the group signature in [11] to secure the communications between vehicles. For the group signature technique, any group member can sign messages on behalf of the group without revealing its real identity. Signatures can be verified using the group public key, thus providing excellent privacy for the users, as the identities of the users are revealed in neither signing nor verifying a message. However, the delay incurred in this technique to verify a signature is linearly proportional to the number of revoked vehicles. Therefore, this technique may not achieve good performance in a large-scale network such as VANETs, where the number of revoked vehicles may be large.

Based on anonymous group signatures, Lu *et al.* [12] propose the efficient conditional privacy preservation (ECPP) protocol for secure vehicular communications, which allows an OBU to get a short-lifetime anonymous certificate from any RSU located in the domain in which the OBU was originally registered. In addition, the certificates of the OBU are free from the key escrow property. The performance of the ECPP protocol is also evaluated under a well-deployed VANET.

Jiang *et al.* [13] develop a verification scheme capable of detecting bogus signatures in batch signature verification schemes, based on a new data structure called a binary authentication tree. In this scheme, a binary tree of the received signatures can be built as follows: 1) The leaf nodes of the tree are the individual signatures, 2) the inner nodes in the level above the leafs are the batch signatures of the leafs directly connected to it, and 3) the upper levels are constructed in the same way as in step 2 until the root of the tree is reached. The verification process is performed in a top-to-bottom manner. At each level of the tree, the batch signature associated with each inner node is verified. If the verification is successfully performed for an inner node x at level $i-1$, this implies that all the signatures located at levels lower than $i-1$ and connected directly or indirectly to the inner node x are correct. If the verification fails for the inner node x, all the batch signatures of the inner nodes connected to x and located one level below, i.e., at level $i-2$, must individually be verified. The process is continued until the leafs of the tree are reached, i.e., until all the bogus signatures are found.

Different from the aforementioned works, we propose an efficient DCS scheme that enables an OBU to update its certificate from any RSU no matter whether the RSU is located in the domain in which the OBU was originally registered or not. Consequently, an OBU is free to roam between domains administered by different authorities. Furthermore, the DCS scheme considers the batch verification of certificates and message signatures. To the best of our knowledge, this is the first approach to address the roaming between different domains in VANETs. Furthermore, the DCS scheme is the first to consider the integration between distributed certificate generation through RSUs and efficient message authentication using batch verification.

TABLE I
NOTATIONS

Symbol	Notation
CA_i, CA_w	two arbitrary CAs
RSU_j, RSU_l	two arbitrary $RSUs$
OBU_m, OBU_n	two arbitrary $OBUs$
s	master secret key of MA for secret key generation
α	partial secret signing-key for signing RSU certificates
γ	partial secret signing-key for signing OBU certificates
P_o	public key used to verify signatures on any message
$S_{\alpha i}$	CA_i secret key to sign RSU certificates
P_α	public key used to verify RSU certificates
$S_{\gamma j_i}$	RSU_j secret key, generated by CA_i, to sign OBU certificates
P_γ	public key used to verify OBU certificates
P_μ	public key used to verify any certificate
PK_i	public key for CA_i
SK_i	secret key for CA_i
PK_{j_i}	RSU_j public key generated by CA_i
SK_{j_i}	RSU_j secret key generated by CA_i
$cert_{RSU_{ji}}$	certificate for RSU_j generated by CA_i
$PK_{m_{ji}}$	OBU_m public key generated by RSU_j using PK_{j_i}
$SK_{m_{ji}}$	OBU_m secret key generated by RSU_j using SK_{j_i}
$vepriod$	OBU certificate validity period
$cert_{OBU_{mji}}$	OBU_m certificate generated by RSU_j using $S_{\gamma j_i}$
t_{stamp}	time stamp
H_1	hash function such that $\{0,1\}^* \in \mathbb{G}_1^*$
H_2	hash function such that $\{0,1\}^* \in \mathbb{Z}_q^*$

III. PRELIMINARIES

In this section, we introduce the bilinear pairings. The notations used throughout this paper are given in Table I.

A. Bilinear Pairing

The bilinear pairing [14] is the foundation of the proposed DCS scheme. Let $\mathbb{G}_1$ denote an additive group of prime order q and $\mathbb{G}_2$ be a multiplicative group of the same order. Let P be a generator of $\mathbb{G}_1$ and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear mapping with the following properties.

1) Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, for all $P, Q \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q$.
2) Nondegeneracy: $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.
3) Symmetric: $\hat{e}(P, Q) = \hat{e}(Q, P)$, for all $P, Q \in \mathbb{G}_1$.
4) Admissible: The map $\hat{e}$ is efficiently computable.

The bilinear map e can be implemented using the Weil [15] and Tate [16] pairings on elliptic curves. We consider the implementation of a Tate pairing on an Miyaji–Nakabayashi–Takano (MNT) curve [17] with embedding degree 6, where $\mathbb{G}_1$ is represented by 161 bits, and the order q is represented by 160 bits. The group order of $\mathbb{G}_1$ is defined as the number of the points on the employed elliptic curve. For an MNT elliptic curve with embedding degree 6 and order q represented by 160 bits, the group order of $\mathbb{G}_1$ is 4.5×10^{30},[1] which qualifies the bilinear pairing as a practical choice for securing the large-scale VANETs.

[1]This result is obtained using the Multiprecision Integer and Rational Arithmetic C/C++ library [18].

The security of the proposed scheme depends on solving the following hard computational problems.

1) *Elliptic-curve discrete-logarithm problem (ECDLP)*: Given a point P of order q on an elliptic curve and a point Q on the same curve, the ECDLP problem [19] is to determine the integer l, $0 \leq l \leq q-1$ such that $Q = lP$.
2) *Computational Diffie–Hellman (CDH) problem*: For two unknowns $a, b \in \mathbb{Z}_p^*$, the CDH problem [20] is the following: Given $aP, bP \in \mathbb{G}_1$, compute $abP \in \mathbb{G}_1$.

IV. SYSTEM DESIGN CONSIDERATIONS IN THE PROPOSED DCS SCHEME

In this section, we discuss the security objectives, system architecture, and network model of the proposed DCS scheme.

A. Security Objectives

In the DCS scheme, we aim to achieve the following security objectives.

1) Authentication: Entity authentication is required to prevent illegitimate users from injecting bogus messages into the network. Each vehicle in the network should possess an authentic identity. When a vehicle receives a message, it first checks the authenticity of the sender identity before performing further processing to the received message. In addition to entity authentication, data authentication is a concern to ensure that the contents of the received data are neither altered nor replayed.
2) Nonrepudiation: Nonrepudiation is necessary to prevent legitimate users from denying the transmission or the content of their messages. Users anticipate the network to provide a high level of liability, where a vehicle involved in a crash should efficiently be identified. Liability can be achieved by investigating the messages saved in each vehicle involved in the crash. However, if nonrepudiation cannot be guaranteed, this process will be trivial.
3) Privacy: Providing privacy is mainly related to preventing the disclosure of the real identity of the users and their location information. Privacy can be provided by introducing identity anonymity such that any observer could neither identify the real identity nor correlate the real identity with the current location of any user. An observer is an attacker launching tracking attacks by installing receivers on the roads to eavesdrop the messages broadcast by the OBUs. By trying to correlate some of the broadcast certificates to an OBU, the observer may be able to track that OBU.
4) Transparent roaming: Users will not be satisfied if, upon roaming between different network domains, they have to go to a central location to upload new security materials, e.g., keys, certificates, etc., to be able to use the VANET services. Transparent roaming is needed to ensure seamless operation of the OBUs in VANETs.
5) Access control: Access control is necessary to ensure reliable and secure operation of the system. Any misbehaving entity should be revoked from the network to protect the safety of other legitimate entities in the network. Moreover, any actions taken by that misbehaving entity should be canceled.

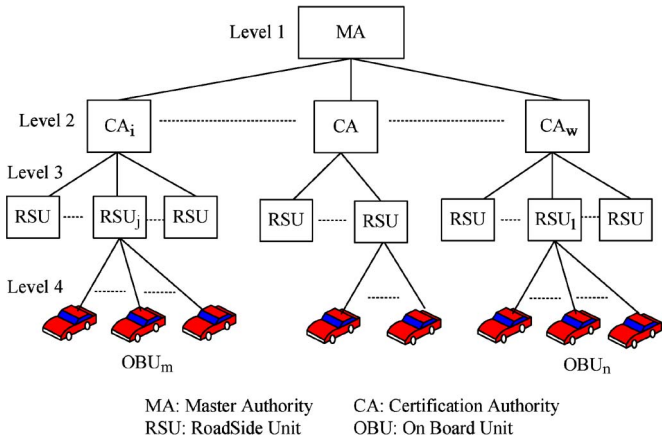


Fig. 1. Proposed DCS hierarchical architecture.

B. Architecture

The DCS hierarchical architecture, which is shown in Fig. 1, consists of four levels: The master authority (MA), which is the root of the system, is located at level 1, the CAs are located at level 2, and the RSUs and the OBUs are located at levels 3 and 4, respectively. In this architecture, entity authentication for RSUs and OBUs is achieved using certificate-based authentication, while that for CAs is achieved using identity-based cryptography [14].

Basic Operation of the DCS Scheme: The basic operation of the DCS scheme is described as follows.

1) The MA is in charge of generating public verification keys for verifying any RSU/OBU certificate. It also generates a public/private key pair for each CA to signal the outgoing messages and verify the incoming messages. Moreover, it generates two secret certificate-signing keys for each CA.
2) A CA uses the first certificate-signing key, which was issued by the MA, to sign a certificate set for each RSU in its coverage area. Each certificate in the RSU certificate set is shared among a group of RSUs. The CA uses the second certificate-signing key as a partial signing key to generate secret OBU-certificate-signing keys for each RSU.
3) An RSU uses the OBU-certificate-signing key to generate short-lifetime anonymous certificates for any OBU. The public verification keys can be used by any entity to verify the certificate of any OBU or RSU, regardless of the issuer of that certificate. This way, any OBU can transparently roam between the coverage areas of different CAs. The certificate generation in DCS is derived from the signature schemes proposed in [21] and [22].

Fig. 2 shows the relations of different keys among the network entities in the DCS scheme.

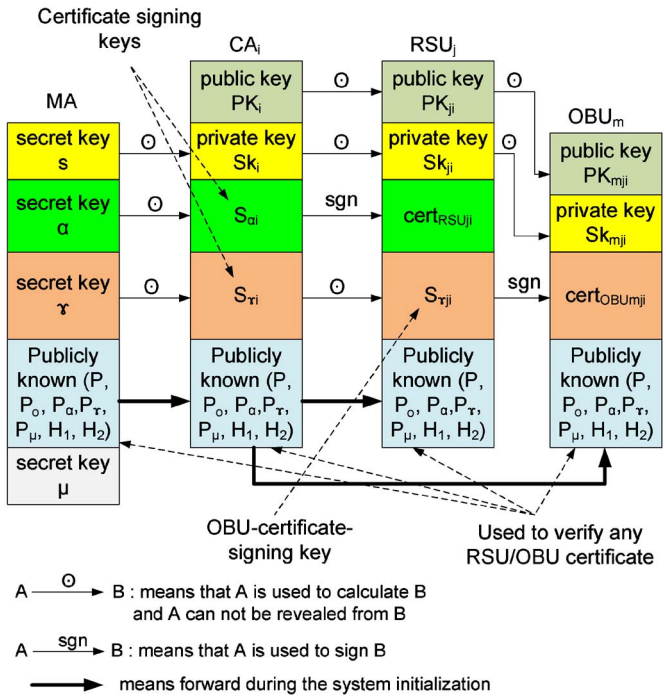


Fig. 2. Relations of different keys among the network entities in the DCS scheme.

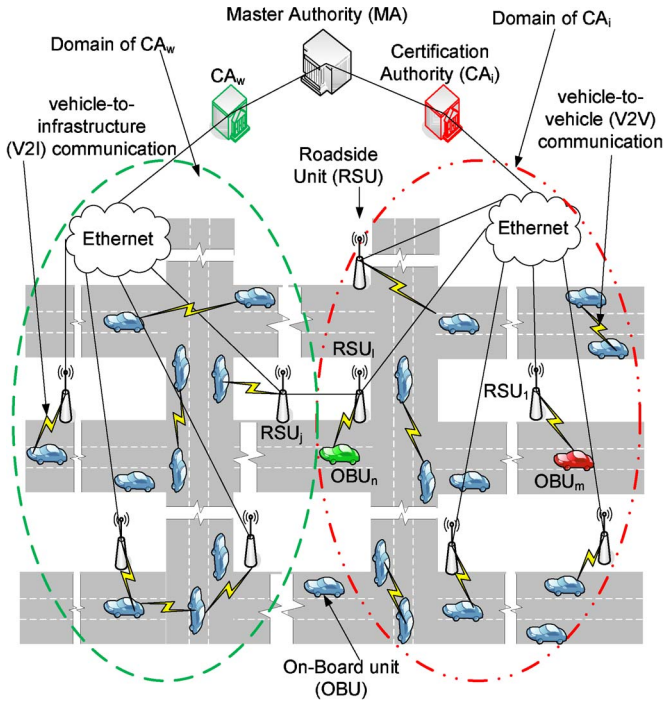


Fig. 3. Network model.

C. Network Model

As shown in Fig. 3, the network model under consideration consists of the following.

1) MA: The MA is the highest level in the system and is trustable by all the network entities. The MA has sufficient physical security measures such that it cannot be compromised, irrespective of the capabilities of an attacker.
2) CAs: Each CA is responsible for generating initial certificates for the RSUs and OBUs in its domain. The CAs are directly connected to the MA. Each CA is physically secure and cannot be compromised.
3) RSUs: RSUs are fixed units distributed in the network. RSUs in one domain are connected via Ethernet to the CA responsible for that domain. Furthermore, an RSU_j at the border of one domain is connected to the nearest RSU_l in an adjacent domain. These connections are required to

check the revocation status of an OBU roaming between two adjacent domains. Moreover, RSUs are responsible for updating the certificates of the OBUs.

4) OBUs: OBUs can communicate with either other OBUs through V2V communications or the infrastructure RSUs through V2I communications. Each OBU is equipped with a GPS receiver, which contains the geographical coordinates of the RSUs. It should be noted that a GPS receiver is necessary for the operation of an OBU in VANETs according to the WAVE standard [8].
5) Hardware security module (HSM): According to the WAVE standard, each network entity is equipped with a tamper-resistant HSM to store its security materials, e.g., secret keys, certificates, etc.

V. Proposed Distributed-Certificate-Service Scheme

In this section, the proposed DCS scheme is presented in detail, including the system initialization, certificate issue, certificate update, and certificate revocation.

A. System Initialization

The initialization stage in the DCS scheme consists of two phases: 1) phase I, which is performed by the MA to generate the security keys necessary for the operation of the DCS scheme and to upload the necessary security keys in the tamper-resistant HSM of each CA, and 2) phase II, which is performed by each CA to upload the required security materials, e.g., keys, certificates, etc., in the tamper-resistant HSM of each OBU and RSU in its domain. It should be noted that both phases of the initialization stage are performed during the registration of CAs with the MA in phase I and RSUs and OBUs with a CA in phase II. In other words, both phases of the initialization stage are performed before triggering any of the VANET services or applications. The details of each phase are given as follows.

1) Phase I: The MA executes Algorithm 1 to generate the necessary secret and public keys for the operation of the DCS scheme and to upload the primary security materials in each CA.

Algorithm 1 Phase I
Require: $\mathrm{ID}_{\mathrm{CA}_i}$
1: Select a random number $s \in \mathbb{Z}_q^*$ as the *master* key, which is part of each entity secret key
2: Set $P_\circ = sP$
3: Select random numbers $\alpha, \gamma \in \mathbb{Z}_q^*$ ▷ *master signing* keys
4: Set $P_\alpha = \alpha P$, $P_\gamma = \gamma P$
5: Select a random number $\mu \in \mathbb{Z}_q^*$
6: Set $P_\mu = \mu P$ ▷ general verification public key
7: Select a hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$
8: Select a hash function $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$
9: **for all** CA_i with identity $\mathrm{ID}_{\mathrm{CA}_i}$ **do**
10: Set $PK_i = Q_i = H_1(\mathrm{ID}_{\mathrm{CA}_i}) \in \mathbb{G}_1^*$ ▷ CA_i public key
11: Set $SK_i = sQ_i$ ▷ CA_i secret key
12: Set $S_{\alpha i} = \alpha Q_i$ ▷ CA_i certificate-signing key
13: Set $S_{\gamma i} = \gamma Q_i$ ▷ CA_i certificate-signing key
14: Upload $SK_i, S_{\alpha i}, S_{\gamma i}, P, P_\circ, P_\alpha, P_\gamma, P_\mu, H_1$, and H_2 in CA_i
15: **end for**

It should be noted that the key s is the *master* secret key, and it is part of the secret key of each entity. Furthermore, the secret keys α and γ are *master signing* keys, and they are parts of each signature on the certificates of the RSUs and OBUs, respectively. Moreover, $P_\circ$, P_α, P_γ, and P_μ are public verification keys, which can be used by any entity in the network to verify any RSU/OBU certificate. In addition, the public key of any CA_i is the hash of its identity $\mathrm{ID}_{\mathrm{CA}_i}$.

By the end of Algorithm 1, each CA has the security materials required to execute phase II.

2) Phase II: In this phase, each CA_i runs Algorithms 2 and 3 to, respectively, initialize each RSU_j and OBU_m in its domain by uploading them with the necessary security materials for their operation in VANETs as follows.

RSU Initialization: Each CA_i executes Algorithm 2 to upload each RSU_j with a certificate $cert_{\mathrm{RSU}_{ji}}$, a secret OBU-certificate-signing key $S_{\gamma j_i}$, which will be used later by RSU_j to issue certificates for OBUs, the minimum and maximum values of the validity period of OBU certificates, and publicly known parameters (P, $P_\circ$, P_α, P_γ, P_μ, H_1, and H_2).

Algorithm 2 Phase II: RSU initialization
Require: $PK_i = Q_i$, $SK_i = sQ_i$, $S_{\alpha i} = \alpha Q_i$, and $S_{\gamma i} = \gamma Q_i$
1: **for all** RSU_j in the domain of CA_i, **do**
2: select random numbers $x_j, a_j \in \mathbb{Z}_q^*$, and a pseudo-identity PID_j for RSU_j
3: Set $SK_{j_i} = x_j SK_i = x_j s Q_i$ ▷ RSU_j secret key
4: Set $PK_{j_i} = x_j PK_i = x_j Q_i$ ▷ RSU_j public key
5: Set $S_{\gamma j_i} = x_j S_{\gamma i} = x_j \gamma Q_i$ ▷ OBU-certificate-signing secret key
6: Set $U_j = a_j P$, $T_j = H_2(PK_{j_i} \| \mathrm{PID}_j \| U_j \| Q_i) \in \mathbb{Z}_q^*$
7: Set $V_j = S_{\alpha i} + a_j T_j P_\mu$
8: Set the certificate of RSU_j as $cert_{\mathrm{RSU}_{ji}} = (PK_{j_i}, U_j, V_j, \mathrm{PID}_j, Q_i)$
9: Select minimum and maximum value for the validity period $(vperiod)$ of any OBU certificate
10: Upload $cert_{\mathrm{RSU}_{ji}}$, $S_{\gamma j_i}$, the minimum and maximum value of $vperiod$, P, $P_\circ$, P_α, P_γ, P_μ, H_1, and H_2 in RSU_j
11: **end for**

Remarks on Algorithm 2:

1) It should be noted that U_j and V_j are the signature of CA_i on $cert_{\mathrm{RSU}_{ji}}$.
2) CA_i stores RSU_j's real identity, PID_j, $cert_{\mathrm{RSU}_{ji}}$, SK_{ji}, and $S_{\gamma ji}$: thus, CA_i can track the operations performed by RSU_j, in case it is compromised, by associating PID_j with its real identity.
3) RSU_j or any other entity can verify the certificate $cert_{\mathrm{RSU}_{ji}}$ by calculating $T_j = H_2(PK_{j_i} \| \mathrm{PID}_j \| U_j \| Q_i)$ and accepting if $\hat{e}(P, V_j) = \hat{e}(P_\alpha, Q_i)\hat{e}(T_j U_j, P_\mu)$. This verification follows since

$$\begin{aligned}\hat{e}(P, V_j) &= \hat{e}(P, S_{\alpha i} + a_j T_j P_\mu) \\ &= \hat{e}(P, \alpha Q_i + a_j T_j P_\mu) \\ &= \hat{e}(P, \alpha Q_i)\hat{e}(P, a_j T_j P_\mu) \\ &= \hat{e}(\alpha P, Q_i)\hat{e}(T_j a_j P, P_\mu) \\ &= \hat{e}(P_\alpha, Q_i)\hat{e}(T_j U_j, P_\mu). \end{aligned} \quad (1)$$

4) The CA repeatedly runs Algorithm 2 to load each RSU with a set of certificates. Each certificate is shared with a different group of RSUs to enforce the anonymous group signature when generating OBU certificates.

OBU initialization: Each CA_i executes Algorithm 3 to upload each OBU_m having identity ID_{OBU_m} in its domain with a number (N_{cert}) of short-lifetime certificates. The identity ID_{OBU_m} is a unique identity loaded in OBU_m during the manufacturing process.

Algorithm 3 Phase II: OBU initialization

Require $\{cert_{\text{RSU}_{ji}} = (PK_{j_i}, U_j, V_j, \text{PID}_j, Q_i), SK_{j_i} = x_j sQ_i$, and $S_{\gamma j_i} = x_j \gamma Q_i\}$ of RSU_j and ID_{OBU_m} of OBU_m

1: **for all** OBU_m in the domain of CA_i **do**
2: Check the validity of ID_{OBU_m}
3: **if** ID_{OBU_m} is invalid **then**
4: **return** $\perp$
5: **else**
6: **for** $r \leftarrow 1$ **to** N_{cert}, CA_i **do**
7: Select random numbers $y_{m,r}, b_{m,r} \in \mathbb{Z}_q^*$
8: Set $y_{m,r} SK_{j_i} = y_{m,r} x_j sQ_i$ ▷ partial secret key
9: Set $y_{m,r} PK_{j_i} = y_{m,r} x_j Q_i$ ▷ partial public key
10: **end for**
11: **return** $\{y_{m,r} SK_{j_i}, y_{m,r} PK_{j_i} | 1 \le r \le N_{\text{cert}}\}$ to OBU_m
12: **for** $r \leftarrow 1$ **to** N_{cert}, OBU_m
13: Select a random number $z_{m,r} \in \mathbb{Z}_q^*$
14: Set the final secret key as

$$SK_{m_{ji},r} = z_{m,r} y_{m,r} SK_{j_i} = z_{m,r} y_{m,r} x_j sQ_i$$

15: Set the final public key as

$$PK_{m_{ji},r} = z_{m,r} y_{m,r} PK_{j_i} = z_{m,r} y_{m,r} x_j Q_i$$

16: **end for**
17: **return** $\{PK_{m_{ji},r} | 1 \le r \le N_{\text{cert}}\}$ to CA_i
18: **for** $r \leftarrow 1$ **to** N_{cert}, CA_i **do**
19: Select a validity period $vperiod_{m,r}$, and a pseudoidentity $\text{PID}_{m,r}$
20: Set $U'_{m,r} = b_{m,r} P$
21: Set $L_{m,r} = H_2(PK_{m_{ji},r} \| vperiod_{m,r} \| \text{PID}_{m,r} \| U'_{m,r}) \in \mathbb{Z}_q^*$
22: Set $V'_{m,r} = S_{\gamma j_i} + b_{m,r} L_{m,r} P_\mu$
23: Set $cert_{\text{OBU}_{mji,r}} = (PK_{m_{ji},r}, U'_{m,r}, V'_{m,r}, vperiod_{m,r}, \text{PID}_{m,r}, cert_{\text{RSU}_{ji}})$
24: **end for**
25: Upload $\{cert_{\text{OBU}_{mji,r}} | 1 \le r \le N_{\text{cert}}\} = \{PK_{m_{ji},r}, U'_{m,r}, V'_{m,r}, vperiod_{m,r}, \text{PID}_{m,r}, cert_{\text{RSU}_{ji}} | 1 \le r \le N_{\text{cert}}\}$, P, $P_\circ$, P_α, P_γ, P_μ, H_1, and H_2 in OBU_m
26: CA_i stores ID_{OBU_m} and $\{\text{PID}_{m,r}, cert_{m_{ji},r}, SK_{m_{ji},r} | 1 \le r \le N_{\text{cert}}\}$
27: **end if**
28: **end for**

Remarks on Algorithm 3:

1) In Algorithm 3, CA_i selects an arbitrary RSU_j in its service area as the certificate issuer and uses the security materials $\{cert_{\text{RSU}_{ji}} = (PK_{j_i}, U_j, V_j, \text{PID}_j, Q_i), SK_{j_i} = x_j sQ_i, S_{\gamma j_i} = x_j \gamma Q_i\}$ of RSU_j. Note that CA_i is the entity that issued these security materials for RSU_j.
2) CA_i stores the real identity $(\text{ID}_{\text{OBU}_m})$ and $\{\text{PID}_{m,r}, cert_{m_{ji},r}, SK_{m_{ji},r} | 1 \le r \le N_{\text{cert}}\}$ of OBU_m; thus, CA_i can efficiently track OBU_m, in case it is compromised, by associating PID_m to ID_{OBU_m}.
3) It should be noted that, throughout the rest of this paper, whenever the subscript r equals 1, it will be omitted for the ease of presentation.
4) Any entity in the network can verify a single certificate $cert_{\text{OBU}_{mji}}$ by verifying $cert_{\text{RSU}_{ji}}$ and then verifying $cert_{\text{OBU}_{mji}}$. Alternatively, $cert_{\text{RSU}_{ji}}$ and $cert_{\text{OBU}_{mji}}$ can aggregately be verified as follows.
   a) Check $vperiod$ and proceed only if it is valid.
   b) Calculate $T_j = H_2(PK_{j_i} \| \text{PID}_j \| U_j \| Q_i)$ and $L_m = H_2(PK_{m_{ji}} \| vperiod \| \text{PID}_m \| U'_m)$.
   c) Accept if $\hat{e}(P, V_j + V'_m) = \hat{e}(P_\alpha, Q_i) \hat{e}(P_\gamma, PK_{j_i}) \hat{e}(T_j U_j + L_m U'_m, P_\mu)$. This verification follows since

$$\begin{aligned}
&\hat{e}\left(P, V_j + V'_m\right) \\
&= \hat{e}(P, S_{\alpha i} + a_j T_j P_\mu + S_{\gamma j_i} + b_m L_m P_\mu) \\
&= \hat{e}(P, \alpha Q_i) \hat{e}(P, x_j \gamma Q_i) \hat{e}(P, a_j T_j P_\mu + b_m L_m P_\mu) \\
&= \hat{e}(\alpha P, Q_i) \hat{e}(\gamma P, x_j Q_i) \hat{e}(a_j T_j P + L_m b_m P, P_\mu) \\
&= \hat{e}(P_\alpha, Q_i) \hat{e}(P_\gamma, PK_{j_i}) \hat{e}\left(T_j U_j + L_m U'_m, P_\mu\right). \quad (2)
\end{aligned}$$

5) Including $cert_{\text{RSU}_{ji}}$ in $cert_{\text{OBU}_{mji}}$ guarantees that $cert_{\text{OBU}_{mji}}$ is generated by a legitimate RSU_j with a valid public key PK_{j_i}. This inclusion also gives the CA the ability to revoke any operation performed by a compromised RSU during the period from the RSU compromising until the detection of the compromised RSU. In other words, consider that an attacker compromises an RSU_l having a certificate $cert_{\text{RSU}_{li}}$ and that the attacker generates some OBU certificates from the compromised RSU_l. When the CA detects that RSU_l is compromised, it revokes $cert_{\text{RSU}_{li}}$. The revocation of $cert_{\text{RSU}_{li}}$ automatically revokes all the OBU certificates generated by RSU_j, as those certificates contain the revoked $cert_{\text{RSU}_{li}}$.

B. OBU Certificates Update

The DCS scheme enables an OBU to update its certificate from an RSU. Thus, the scalability of the DCS scheme stems from the distributed certification service compared with the centralized certification service in the classical PKI, where an OBU has to contact a CA to update its certificate. Since the DCS scheme depends on the RSUs to update the certificates of the OBUs, the density of RSUs is crucial to the operation of the DCS scheme. In this section, we discuss the adaptability of the DCS scheme to different densities of RSUs and how an OBU can dynamically update its certificates, even if it is roaming

between different domains. In the certificate update process, an RSU generates a number of short-lifetime anonymous certificates for an OBU sufficient for securing the communications of the OBU until it meets another RSU. The number of generated certificates by an RSU depends on the RSU density.

1) Adapting DCS to Different RSU Densities: In this section, we discuss how the DCS scheme can adapt to different densities of RSUs. Let T_{RSU} denote the duration an OBU spent between meeting two different RSUs on its way. When the number of RSUs in a given area increases, it is intuitive that T_{RSU} will decrease and *vice versa*, i.e., T_{RSU} is inversely proportional to the RSU density. It should be noted that an OBU has to periodically change its certificate during T_{RSU} to avoid being tracked. Since an OBU spends a time of $vperiod$, which is the validity period of the OBU certificate, using the same certificate, the number of certificates N_{cert} required to protect the privacy of that OBU in the duration it spent between meeting two different RSUs can be calculated as follows:

$$N_{\mathrm{cert}} = \left\lceil \frac{T_{\mathrm{RSU}}}{vperiod} \right\rceil. \tag{3}$$

An OBU_n moving on the road can calculate its T_{RSU} value based on its direction and speed and the coordinates of the RSUs initially loaded in its GPS receiver. When OBU_n needs to update its certificates, it sends a request to update its certificate and the value of its T_{RSU} to an RSU_j. Then, using (3) and the appropriate value for $vperiod$, RSU_j can calculate the required number of certificates (N_{cert}) that should be generated to the requesting OBU_n to protect its privacy until it meets the next RSU on its way. This way, the DCS scheme can adapt to different RSU densities.

2) OBU Dynamic Certificate Update: The DCS offers full interoperability for any OBU to update its certificate in a completely transparent way, even when it roams in a domain different from its home domain. Consider that OBU_n, with certificate $cert_{\mathrm{OBU}_{nlw}} = (PK_{n_{lw}}, U_n^{\backprime}, V_n^{\backprime}, vperiod, \mathrm{PID}_n, cert_{\mathrm{RSU}_{lw}})$ generated by RSU_l in the domain of CA_w, enters the domain of CA_i and needs to update its certificate from RSU_j, which has a certificate $cert_{\mathrm{RSU}_{ji}} = (PK_{j_i}, U_j, V_j, \mathrm{PID}_j, Q_i)$, as shown in Fig. 3, where OBU_n is shown in green. The certificate update algorithm, which is shown in Fig. 4, has two phases: *phase I* for mutual authentication and generating a shared secret key in a noninteractive way and *phase II* to issue a bundle of N_{cert} short-lifetime anonymous certificates for OBU_n. The *OBU-certificate-update* algorithm is described as follows.

Phase I:

1) When OBU_n receives the periodically broadcast certificate $cert_{\mathrm{RSU}_{ji}}$ of RSU_j, it verifies $cert_{\mathrm{RSU}_{ji}}$ by calculating $T_j = H_2(PK_{j_i} \| \mathrm{PID}_j \| U_j \| Q_i)$ and proceeds only if $\hat{e}(P, V_j) = \hat{e}(P_\alpha, Q_i)\hat{e}(T_j U_j, P_\mu)$. If valid, OBU_n calculates the shared secret key (k_{nj}) using its secret key $SK_{n_{lw}}$ and the public key PK_{j_i} of RSU_j included in $cert_{\mathrm{RSU}_{ji}}$ as $k_{nj} = \hat{e}(SK_{n_{lw}}, PK_{j_i}) = \hat{e}(z_n y_n x_l s Q_w, x_j Q_i) = \hat{e}(Q_w, Q_i)^{z_n y_n x_l x_j s} = k_{jn}$. Then, OBU_n calculates T_{RSU} based on its speed and destination and the loaded coordinates of the RSUs. After that, OBU_n encrypts T_{RSU} with k_{nj} and sends its certificate $cert_{\mathrm{OBU}_{nlw}}$ along with the encrypted T_{RSU} to RSU_j.

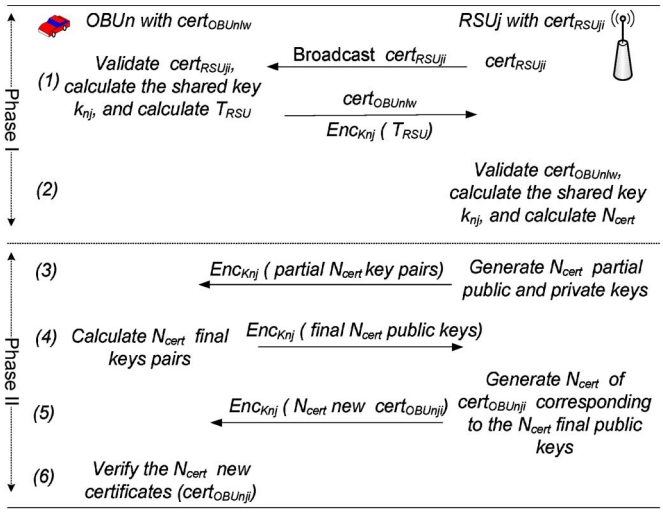


Fig. 4. Onboard unit certificate update.

2) RSU_j verifies $cert_{\mathrm{OBU}_{nlw}}$ by calculating $T_l = H_2(PK_{l_w} \| \mathrm{PID}_l \| U_l \| Q_w)$ and $L_n = H_2(PK_{n_{lw}} \| vperiod \| \mathrm{PID}_n \| U_n^{\backprime})$ and proceeds only if $\hat{e}(P, V_l + V_n^{\backprime}) = \hat{e}(P_\alpha, Q_l)\hat{e}(P_\gamma, PK_{l_w})\hat{e}(T_l U_l + L_n U_n^{\backprime}, P_\mu)$. If valid, RSU_j calculates the shared secret key as $k_{jn} = \hat{e}(PK_{n_{lw}}, SK_{j_i}) = \hat{e}(z_n y_n x_l Q_w, x_j s Q_i) = \hat{e}(Q_w, Q_i)^{z_n y_n x_l x_j s} = k_{nj}$ in a noninteractive key agreement way. Then, RSU_j decrypts T_{RSU} using k_{nj} and calculates N_{cert} using (3) based on the bounds of the certificate validity period $vperiod$ set by CA_i.

Phase II:

1) As shown in Fig. 4, RSU_j selects N_{cert} random numbers $\{y_{n,r}^{\backprime} | 1 \le r \le N_{\mathrm{cert}}\} \in \mathbb{Z}_q^*$ and calculates N_{cert} partial secret keys as $\{y_{n,r}^{\backprime} x_j s Q_i | 1 \le r \le N_{\mathrm{cert}}\}$ and the corresponding N_{cert} partial public keys $\{y_{n,r}^{\backprime} x_j Q_i | 1 \le r \le N_{\mathrm{cert}}\}$. Then, it securely delivers the partial key pairs to OBU_n by encrypting them with the shared secret key k_{nj} established in *phase I*.
2) OBU_n selects N_{cert} random numbers $\{z_{n,r}^{\backprime} \in \mathbb{Z}_q^* | 1 \le r \le N_{\mathrm{cert}}\}$ and calculates its final secret keys $\{SK_{n_{ji},r} | 1 \le r \le N_{\mathrm{cert}}\} = \{z_{n,r}^{\backprime} y_{n,r}^{\backprime} x_j s Q_i | 1 \le r \le N_{\mathrm{cert}}\}$ and its final public key $\{PK_{n_{ji},r} | 1 \le r \le N_{\mathrm{cert}}\} = \{z_{n,r}^{\backprime} y_{n,r}^{\backprime} x_j Q_i | 1 \le r \le N_{\mathrm{cert}}\}$. After that, OBU_n sends its final public keys $\{PK_{n_{ji},r} | 1 \le r \le N_{\mathrm{cert}}\}$ to RSU_j.
3) For each key in $\{PK_{n_{ji},r} | 1 \le r \le N_{\mathrm{cert}}\}$, RSU_j chooses a validity period $vperiod_{n,r}$ and a pseudoidentity $\mathrm{PID}_{n,r}$. After that, RSU_j selects N_{cert} random numbers $\{b_{n,r}^{\backprime} | 1 \le r \le N_{\mathrm{cert}}\} \in \mathbb{Z}_q^*$ and calculates $\{U_{n,r}^{\backprime} | 1 \le r \le N_{\mathrm{cert}}\} = \{b_{n,r}^{\backprime} P | 1 \le r \le N_{\mathrm{cert}}\}$, $\{L_{n,r}^{\backprime} | 1 \le r \le N_{\mathrm{cert}}\} = \{H_2(PK_{n_{ji},r} \| vperiod_{n,r} \| \mathrm{PID}_{n,r} \| U_{n,r}^{\backprime}) | 1 \le r \le N_{\mathrm{cert}}\} \in \mathbb{Z}_q^*$, and $\{V_{n,r}^{\backprime} | 1 \le r \le N_{\mathrm{cert}}\} = \{S_{\gamma j_i} + b_{n,r}^{\backprime} L_{n,r}^{\backprime} P_\mu | 1 \le r \le N_{\mathrm{cert}}\}$. Finally, RSU_j issues $\{cert_{\mathrm{OBU}_{nji,r}} | 1 \le r \le N_{\mathrm{cert}}\} = \{(PK_{n_{ji},r}, U_{n,r}^{\backprime}, V_{n,r}^{\backprime}, vperiod_{n,r}, \mathrm{PID}_{n,r}, cert_{\mathrm{RSU}_{ji}}) | 1 \le r \le N_{\mathrm{cert}}\}$ and delivers them to OBU_n over a channel secured by the key k_{nj}.
4) OBU_n verifies the received certificates $\{cert_{\mathrm{OBU}_{nji,r}} | 1 \le r \le N_{\mathrm{cert}}\}$ by calculating $\{L_{n,r}^{\backprime} | 1 \le r \le N_{\mathrm{cert}}\} =$

$\{H_2(PK_{n_{ji}}|vperiod_{n,r}\|\mathrm{PID}_{n,r}\|U^{\backprime}_{n,r})\|1 \le r \le N_{\mathrm{cert}}\}$ and accepts only if

$$\hat{e}\left(P, \sum_{r=1}^{N_{\mathrm{cert}}} V^{\backprime}_{n,r}\right) = \hat{e}\left(P_\gamma, \sum_{r=1}^{N_{\mathrm{cert}}} PK_{j_i}\right)\hat{e}\left(\sum_{r=1}^{N_{\mathrm{cert}}} L^{\backprime}_{n,r}U^{\backprime r}_{n}, P_\mu\right). \tag{4}$$

This verification holds since

$$\begin{aligned}
&\hat{e}\left(P, \sum_{r=1}^{N_{\mathrm{cert}}} V^{\backprime}_{n,r}\right)\\
&= \hat{e}\left(P, V^{\backprime}_{n,1} + V^{\backprime}_{n,2} + \cdots + V^{\backprime}_{n,N_{\mathrm{cert}}}\right)\\
&= \hat{e}\left(P, S_{\gamma j_i} + b^{\backprime}_{n,1}L^{\backprime}_{n,1}P_\mu + S_{\gamma j_i} + b^{\backprime}_{n,2}L^{\backprime}_{n,2}P_\mu + \cdots\right.\\
&\qquad \left. + S_{\gamma j_i} + b^{\backprime}_{n,N_{\mathrm{cert}}}L^{\backprime}_{n,N_{\mathrm{cert}}}P_\mu\right)\\
&= \hat{e}\left(P, \sum_{r=1}^{N_{\mathrm{cert}}} S_{\gamma j_i}\right)\hat{e}\left(P, b^{\backprime}_{n,1}L^{\backprime}_{n,1}P_\mu + b^{\backprime}_{n,2}L^{\backprime}_{n,2}P_\mu + \cdots\right.\\
&\qquad \left. + b^{\backprime}_{n,N_{\mathrm{cert}}}L^{\backprime}_{n,N_{\mathrm{cert}}}P_\mu\right)\\
&= \hat{e}\left(P, \sum_{r=1}^{N_{\mathrm{cert}}} x_j\gamma Q_i\right)\hat{e}\left(L^{\backprime}_{n,1}b^{\backprime}_{n,1}P + L^{\backprime}_{n,2}b^{\backprime}_{n,2} + \cdots\right.\\
&\qquad \left. + L^{\backprime}_{n,N_{\mathrm{cert}}}b^{\backprime}_{n,N_{\mathrm{cert}}}P, P_\mu\right)\\
&= \hat{e}\left(\gamma P, \sum_{r=1}^{N_{\mathrm{cert}}} x_j Q_i\right)\hat{e}\left(L^{\backprime}_{n,1}U^{\backprime}_{n,1} + L^{\backprime}_{n,2}U^{\backprime}_{n,2} + \cdots\right.\\
&\qquad \left. + L^{\backprime}_{n,N_{\mathrm{cert}}}U^{\backprime}_{n,N_{\mathrm{cert}}}, P_\mu\right)\\
&= \hat{e}\left(P_\gamma, \sum_{r=1}^{N_{\mathrm{cert}}} PK_{j_i}\right)\hat{e}\left(\sum_{r=1}^{N_{\mathrm{cert}}} L^{\backprime}_{n,r}U^{\backprime}_{n,r}, P_\mu\right).
\end{aligned} \tag{5}$$

By the end of *phase II*, OBU_n gets N_{cert} short-lifetime anonymous certificates, which are sufficient to protect its privacy until it meets another RSU on its way.

Remarks:

1) The preceding algorithm enables an OBU_m from one domain (CA_w) to securely update its certificate in another domain (CA_i). In particular, if $i = w$, OBU_m updates its certification in its local domain.
2) By increasing the number of the short-lifetime certificates an OBU can get from an RSU, the distance an OBU can move without the need to contact another RSU to update its certificates increases. In other words, by changing the number of certificates N_{cert}, the DCS scheme can adapt to different densities of RSUs. Consider a constant $vperiod = 1$ min [1] for all the certificates of an OBU, and the OBU average speed in a domain is 60 km/h. When an OBU updates its certificates from an RSU for values of N_{cert} equal five and ten certificates, these values are sufficient to protect the privacy of that OBU over distances of 5 and 10 km, respectively, without the need to contact another RSU.
3) When an RSU_j uses one of its certificates $(cert_{\mathrm{RSU}_{ji}})$ and signing keys $(S_{\gamma j_i})$ to issue a certificate for an OBU, this corresponds to using anonymous group signatures since $S_{\gamma j_i}$ and $cert_{\mathrm{RSU}_{ji}}$ are shared among multiple RSUs. Furthermore, the generated certificate for OBU contains a pseudoidentity, which cannot be related to the real identity of the OBU. Since an observer can link an OBU certificate to neither the real identity of the OBU nor the location of the RSU which issued that certificate, the issued certificate $cert_{\mathrm{OBU}_{nji}}$ is anonymous.
4) The noninteractive key agreement in *phase I* (steps 1 and 2) is very attractive to vehicular networks, since it enables any entity $\mathcal{A}$ to establish a shared secret key with another entity $\mathcal{B}$ by calculating the bilinear pairing of its secret key and the public key of $\mathcal{B}$. The noninteractive key agreement is of significant importance to update certificates and establishing secure channels in VANETs.

C. Certificate Revocation

Revocation is required to prevent compromised entities from accessing the network. In the DCS scheme, we adopt the certificate-revocation-list (CRL) method, which is the revocation method employed in the WAVE standard [8]. A CRL is a list containing all the identities and the validity periods of the revoked certificates. It should be noted that the short-lifetime certificates of OBUs will be self-revoked after their lifetime expires. The certificates of an entity (OBU or RSU) are added to a CRL only if the entity is compromised. When an entity (OBU or RSU) is compromised in one domain, the CA responsible for that domain adds all the certificates of the compromised entity to the current CRL and broadcasts the new CRL in its domain. Each entity continuously maintains the recently received CRL by removing the certificates with expired validity periods.

According to the distribution of the CRLs in the DCS scheme, each CA distributes the CRL to the RSUs in its domain through its local Ethernet. Then, the RSUs receiving the new CRL broadcasts it to all the OBUs in that domain. Furthermore, the CRL is delivered from the border RSUs in one domain (i) to the border RSUs in the adjacent domain (w) to enable the RSUs in domain (w) to check the revocation status of the OBUs coming from domain (i). However, the CRL corresponding to domain (i) will be kept in the border RSUs in domain w, and it will not be further broadcast in domain w. For example, a CRL is broadcast by CA_i in its domain (see Fig. 3). This CRL is broadcast in domain i until it reaches RSU_l. Then, RSU_l broadcasts this CRL in its coverage area, and it delivers this CRL to the RSU_j in domain w. RSU_j stores this CRL to check the revocation status of the OBUs moving from domain i to domain w. In the case RSUs do not completely cover the domain of a CA, Laberteaux *et al.* [23] show that V2V communication can be used to efficiently distribute a CRL to all the OBUs. More results about the efficiency of the CRL distribution using V2V communications can be found in [23].

VI. Certificate-Based Message Signature and Verification

To satisfy the data authentication and nonrepudiation security requirements of VANETs, each entity in the system should be capable of signing and verifying a given message with the corresponding certificate. In this section, we present the basic message signature and verification, followed by the proposed batch verification for message signatures and certificates.

A. OBU/RSU/CA Message Signature and Verification

An OBU_m with $cert_{\text{OBU}_{mji}}$ can generate a valid signature $(U_m^{\backslash\backslash\backslash}, V_m^{\backslash\backslash\backslash})$ for a given message M as follows.

1) Select a random number $c_m \in \mathbb{Z}_q^*$.
2) Calculate $U_m^{\backslash\backslash\backslash}$, R_m, and $V_m^{\backslash\backslash\backslash}$, where $U_m^{\backslash\backslash\backslash} = c_m P$, $R_m = H_2(M \| PK_{m_{ji}} \| U_m^{\backslash\backslash\backslash} \| \text{PID}_m \| t_{\text{stamp}}) \in \mathbb{Z}_q^*$, and $V_m^{\backslash\backslash\backslash} = SK_{m_{ji}} + c_m R_m P_\mu$.
3) $(U_m^{\backslash\backslash\backslash}, V_m^{\backslash\backslash\backslash})$ is a valid signature on M.

Any entity in the network can verify the signature $(U_m^{\backslash\backslash\backslash}, V_m^{\backslash\backslash\backslash})$ on the message M as follows.

1) Verify that the sender of the message is a valid user and check the time stamp t_{stamp}.
2) Calculate

$$R_m = H_2\left(M \left\| PK_{m_{ji}} \right\| U_m^{\backslash\backslash\backslash} \| \text{PID}_m \| t_{\text{stamp}}\right). \quad (6)$$

3) Accept if

$$\begin{aligned}\hat{e}\left(P, V_m^{\backslash\backslash\backslash}\right) &= \hat{e}(P, SK_{m_{ji}} + c_m R_m P_\mu) \\ &= \hat{e}(P, SK_{m_{ji}})\hat{e}(P, c_m R_m P_\mu) \\ &= \hat{e}(P, z_m y_m x_j s Q_i)\hat{e}(P, c_m R_m P_\mu) \\ &= \hat{e}(sP, z_m y_m x_j Q_i)\hat{e}(R_m c_m P, P_\mu) \\ &= \hat{e}(PK_{m_{ji}}, P_\circ)\hat{e}\left(R_m U_m^{\backslash\backslash\backslash}, P_\mu\right). \end{aligned} \quad (7)$$

Similarly, any CA or RSU can sign an arbitrary message using the aforementioned procedures.

B. Batch Verification for Message Signatures

Consider that an OBU $\mathcal{A}$ receives $(U_1^{\backslash\backslash\backslash}, V_1^{\backslash\backslash\backslash}), (U_2^{\backslash\backslash\backslash}, V_2^{\backslash\backslash\backslash}), \ldots, (U_K^{\backslash\backslash\backslash}, V_K^{\backslash\backslash\backslash})$, which are the signatures on the messages $M_1, M_2, \ldots, M_K$, respectively. Then, those signatures can aggregately be verified as follows.

1) Calculate $\overline{V}^{\backslash\backslash\backslash} = \sum_{k=1}^{K} V_k^{\backslash\backslash\backslash}$ and $R_1, R_2, \ldots, R_K$ as in (6).
2) Calculate $\overline{U}^{\backslash\backslash\backslash} = \sum_{k=1}^{K} R_k U_k^{\backslash\backslash\backslash}$.
3) Accept if

$$\hat{e}(P, \overline{V}^{\backslash\backslash\backslash}) = \hat{e}\left(P_\circ, \sum_{k=1}^{K} PK_{\text{OBU},k}\right)\hat{e}(\overline{U}^{\backslash\backslash\backslash}, P_\mu) \quad (8)$$

where $PK_{\text{OBU},k}$ is the public key in certificate k.

Proof: First, we consider an OBU $\mathcal{A}$ receives two messages from OBU_m and OBU_n, where OBU_m generates a signature $(U_m^{\backslash\backslash\backslash}, V_m^{\backslash\backslash\backslash})$ on the message M_1, where $U_m^{\backslash\backslash\backslash} = c_m P$, and $V_m^{\backslash\backslash\backslash} = SK_m + c_m R_m P_\mu = z_m y_m x_j s Q_i + c_m R_m P_\mu$. In addition, OBU_n generates a signature $(U_n^{\backslash\backslash\backslash}, V_n^{\backslash\backslash\backslash})$ on the message M_2, where $U_n^{\backslash\backslash\backslash} = c_n P$, and $V_n^{\backslash\backslash\backslash} = SK_m + c_n R_n P_\mu = z_n y_n x_j s Q_i + c_n R_n P_\mu$. OBU $\mathcal{A}$ calculates $\overline{V}^{\backslash\backslash\backslash} = V_m^{\backslash\backslash\backslash} + V_n^{\backslash\backslash\backslash} = z_m y_m x_j s Q_i + c_m R_m P_\mu + z_n y_n x_j s Q_i + c_n R_n P_\mu$. The received signatures can aggregately be verified by calculating R_m and R_n and checking that

$$\begin{aligned}&\hat{e}(P, \overline{V}^{\backslash\backslash\backslash}) \\ &= \hat{e}(P, z_m y_m x_j s Q_i + c_m R_m P_\mu + z_n y_n x_j s Q_i + c_n R_n P_\mu) \\ &= \hat{e}(P, z_m y_m x_j s Q_i + z_n y_n x_j s Q_i)\hat{e}(P, c_m R_m P_\mu + c_n R_n P_\mu) \\ &= \hat{e}(sP, z_m y_m x_j Q_i + z_n y_n x_j Q_i)\hat{e}(R_m c_m + P R_n c_n P, P_\mu) \\ &= \hat{e}\left(P_\circ, PK_{m_{ji}} + PK_{n_{ji}}\right)\hat{e}\left(R_m U_m^{\backslash\backslash\backslash} + R_n U_n^{\backslash\backslash\backslash}, P_\mu\right) \\ &= \hat{e}\left(P_\circ, \sum_{k=1}^{2} PK_{\text{OBU},k}\right)\hat{e}(\overline{U}^{\backslash\backslash\backslash}, P_\mu). \end{aligned} \quad (9)$$

As for the multiple messages, they can be verified in a similar way.

C. Batch Verification for Certificates

Consider an OBU_m with certificate $cert_{\text{OBU}_{mji}} = (PK_{m_{ji}}, U_m^{\backslash}, V_m^{\backslash}, vperiod_m, \text{PID}_m, cert_{\text{RSU}_{ji}})$ and an OBU_n with certificate $cert_{\text{OBU}_{nlw}} = (PK_{n_{lw}}, U_n^{\backslash}, V_n^{\backslash}, vperiod_n, \text{PID}_n, cert_{\text{RSU}_{lw}})$, where $cert_{\text{RSU}_{ji}} = (PK_{j_i}, U_j, V_j, \text{PID}_j, Q_i)$, and $cert_{\text{RSU}_{lw}} = (PK_{l_w}, U_l, V_l, \text{PID}_l, Q_w)$. An independent third party can aggregately verify the OBU certificates and the RSU certificates included in them as follows.

1) Check the $vperiod$ of each certificate, and proceed only if it is valid.
2) Calculate $T_j = H_2(PK_{j_i} \| \text{PID}_j \| U_j \| Q_i)$ and $T_l = H_2(PK_{l_i} \| \text{PID}_l \| U_l \| Q_i)$.
3) Calculate $L_m = H_2(PK_{m_{ji}} \| vperiod_m \| \text{PID}_m \| U_m^{\backslash})$ and $L_n = H_2(PK_{n_{lw}} \| vperiod_n \| \text{PID}_n \| U_n^{\backslash})$.
4) Calculate $\overline{V} = V_j + V_l$, $\overline{V}^{\backslash} = V_m^{\backslash} + V_n^{\backslash}$, $\overline{U} = T_j U_j + T_l U_l$, and $\overline{U}^{\backslash} = L_m U^{\backslash} + L_n U^{\backslash}$.
5) Accept if $\hat{e}(P, \overline{V} + \overline{V}^{\backslash}) = (P_\alpha, Q_i + Q_w)\hat{e}(P_\gamma, PK_{ji} + PK_{lw})\hat{e}(\overline{U} + \overline{U}^{\backslash}, P_\mu)$. This verification holds since

$$\begin{aligned}&\hat{e}(P, \overline{V} + \overline{V}^{\backslash}) \\ &= \hat{e}(P, S_{\alpha i} + a_j T_j P_\mu + S_{\alpha w} + a_l T_l P_\mu \\ &\qquad + S_{\gamma ji} + b_m L_m P_\mu + S_{\gamma lw} + b_n L_n P_\mu) \\ &= \hat{e}(P, \alpha Q_i + a_j T_j P_\mu + \alpha Q_w + a_l T_l P_\mu + x_j \gamma Q_i \\ &\qquad + b_m L_m P_\mu + x_l \gamma Q_w + b_n L_n P_\mu) \\ &= \hat{e}(P, \alpha Q_i + \alpha Q_w)(P, x_j \gamma Q_i + x_l \gamma Q_w) \\ &\quad \times \hat{e}(P, a_j T_j P_\mu + a_l T_l P_\mu + b_m L_m P_\mu + b_n L_n P_\mu) \\ &= \hat{e}(\alpha P, Q_i + Q_w)\hat{e}(\gamma P, x_j Q_i + x_l Q_w) \\ &\quad \times \hat{e}(T_j U_j + T_l U_l + L_m U^{\backslash} + L_n U^{\backslash}, P_\mu) \\ &= \hat{e}(P_\alpha, Q_i + Q_w)\hat{e}(P_\gamma, PK_{ji} + PK_{lw})\hat{e}(\overline{U} + \overline{U}^{\backslash}, P_\mu). \end{aligned} \quad (10)$$

For K OBUs, their certificates can aggregately be verified as follows:

$$\begin{aligned}&\hat{e}(P, \overline{V} + \overline{V}^{\backslash}) \\ &= \hat{e}\left(P_\alpha, \sum_{k=1}^{K} Q_k\right)\hat{e}\left(P_\gamma, \sum_{k=1}^{K} PK_{\text{RSU},k}\right)\hat{e}(\overline{U} + \overline{U}^{\backslash}, P_\mu) \end{aligned} \quad (11)$$

where $\overline{V} = \sum_{k=1}^{K} V_k$, $\overline{V}^{\backslash} = \sum_{k=1}^{K} V_k^{\backslash}$, $PK_{\text{RSU},k} = PK_{ji} + PK_{lw} + \cdots$, $\overline{U} = \sum_{k=1}^{K} T_k U_k$, and $\overline{U}^{\backslash} = \sum_{k=1}^{K} L_k U_k^{\backslash}$.

D. Batch Verification for Message Signatures and Certificates

Consider K OBUs with K certificates generating different K signatures on different K messages. An independent third party can aggregately verify the K signatures and certificates by combining (8) and (11) as follows:

$$\hat{e}(P,\overline{V}+\overline{V}^{\backprime}+\overline{V}^{\backprime\backprime}) = \hat{e}\left(P_{\circ},\sum_{k=1}^{K} PK_{\mathrm{OBU},k}\right)\hat{e}\left(P_{\alpha},\sum_{k=1}^{K} Q_k\right) \times \hat{e}\left(P_{\gamma},\sum_{k=1}^{K} PK_{\mathrm{RSU},k}\right)\hat{e}(\overline{U}+\overline{U}^{\backprime}+\overline{U}^{\backprime\backprime},P_{\mu}). \quad (12)$$

The proof of (12) directly follows from (9) and (10). Equation (12) shows that the DCS scheme overcomes the need to separately verify signatures and certificates of the senders, which is common to most of the existing batch-verification schemes. The DCS scheme amplifies the capabilities of any entity in the network to simultaneously verify a relatively large number of signatures and certificates compared with the conventional verification method, which verifies signatures and certificates one by one, thus decreasing the verification overhead.

It should be noted that (12) can be used by any OBU or RSU to verify the signatures and the certificates included in the different K messages sent by K OBUs. Consequently, (12) represents how authentication can be achieved in V2V and V2I communications.

When there are invalid signatures in the received messages, the data cross-checking technique employed in the WAVE standard can alleviate the effect of the invalid signatures. Specifically, each OBU_n compares the data included in the received message from an OBU_m with those received from other OBUs. If there is a mismatch, OBU_n rejects the message. It should be noted that the data cross-checking technique is useful only when the data contents of the message are malicious. However, if either the data contents of the message are correct and the signature is invalid or the message and signature are correct and the certificate is invalid, this technique is not useful. In such a case, a search approach based on the binary authentication tree [13] can be employed to avoid individually verifying every signature. The basic concept of the binary authentication tree is introduced in Section II. The performance evaluation under such scenario is not trivial [13].

VII. Security Analysis

In this section, we evaluate the proposed DCS scheme according to the security objectives presented in Section IV-A.

1) Authentication: It can be seen that finding the secret keys s, α, γ, and μ from the corresponding public keys $P_{\circ}$, P_{α}, P_{γ}, and P_{μ} are instances of the ECDLP. For example, to find s, we have the following ECDLP: Given P and $P_{\circ}=sP$, find s. In DCS, the authentication of RSUs and OBUs is achieved using digital certificates. For example, the signature of any CA_i on the certificate of any RSU_j is (U_j, V_j), where $U_j = a_jP$, $T_j = H_2(PK_{j_i}\|\mathrm{PID}_j\|U_j\|Q_i) \in \mathbb{Z}_q^*$, and $V_j = S_{\alpha i} + a_jT_jP_{\mu}$. It can be seen that, to forge the certificate of any RSU_j, an attacker should know either $S_{\alpha i} = \alpha Q_i$ or $a_jT_jP_{\mu}$. Since Q_i is publicly known, finding $S_{\alpha i}$ reduces to finding α, which is the ECDLP, as indicated earlier. Furthermore, since T_j can easily be obtained from the certificate of RSU_j, finding $a_jT_jP_{\mu}$ reduces to finding a_jP_{μ}, which can be formulated as a CDH problem, i.e., given $U_j = a_jP$ and $P_{\mu} = \mu P$, find $a_jP_{\mu} = a_j\mu P$. The hardness of the CDH problem is closely related to solving the discrete-logarithm problem [20]. A similar analogy applies to the OBU certificates. Since the ECDLP and the CDH problem are hard computational problems [19], [20], i.e., they cannot be solved in a subexponential time, the certificates of RSUs and OBUs are unforgeable. Since in each communication, an authentication of the sender is performed first, an illegitimate entity cannot communicate with the authentic network users. Furthermore, data authentication is achieved by employing digital signatures, where any message transmitted by any CA, RSU, or OBU has to be signed first. Consequently, any message alteration during the transmission will be detected by the recipient. In clogging attacks, an attacker tries to impersonate a legitimate user and overwhelms legitimate entities in the network by involving them in a large volume of key exchange or by sending bogus messages [24]. In the DCS scheme, each OBU/RSU authenticates the received messages before being involved in any key exchange or responding to the received message. According to [24], since authentication is first done before taking any action, the clogging attacks is hard to launch in the proposed DCS scheme.
2) Nonrepudiation: Nonrepudiation is achieved by requiring all the messages exchanged in the network to be digitally signed by its issuer. For example, the signature of any OBU_m on an arbitrary message M is $(U^{\backprime\backprime}, V^{\backprime\backprime})$, where $U^{\backprime\backprime} = cP$, and $V^{\backprime\backprime} = SK_{m_{ji}} + cRP_{\mu}$. Similar to the aforementioned discussion of the security of RSU certificates, to forge the signature of OBU_m on M, the attacker has to find either $SK_{m_{ji}}$, which is the ECDLP, or cRP_{μ}, which is the CDH problem. Consequently, the signature of any entity cannot be forged. In addition, since nonrepudiation is guaranteed, the liability requirement is also achieved since users cannot deny the transmission or the content of their messages.
3) Privacy: In DCS, privacy is preserved by the following techniques.
   a) *Anonymous authentication:* Anonymous authentication is employed in DCS in the sense that each OBU has a certificate containing only a pseudoidentity, which cannot lead in any way to the real identity of the OBU. Furthermore, by deploying anonymous authentication, the DCS scheme can efficiently prevent an adversary from tracking the real identity of the users.
   b) *Frequent certificate update:* OBU certificates have a short lifetime. As a result, each OBU has to periodically change its certificate, which decreases the probability of being tracked by an external observer.
   c) *Anonymous certificate issuer:* Since each RSU certificate is shared among multiple RSUs, the RSU

certificate included in each OBU certificate cannot lead to the location where the OBU issued its certificate.

d) *Avoiding key escrow:* When an OBU_m updates its certificate from an RSU_j, RSU_j sends a partial secret key $y_m x_j s Q_i$ to OBU_m. After that, OBU_m calculates its final secret key as $SK_{m_{ji}} = z_m y_m x_j s Q_i$. It can be seen that finding $SK_{m_{ji}}$ from the partial secret key is the ECDLP. Since the secret key of any OBU cannot be forged, the DCS is free from the key screw, which is common to any PKI. As a result, the messages signed by the secret key of any OBU can only be verified by the public key of that OBU, and this signature cannot be generated by any other entity in the network, hence achieving a high privacy level.

Although the DCS offers a coalition of privacy-preserving mechanisms, an observer can still launch a tracking attack on an OBU. However, this tracking attack requires an observer to launch a large number of receivers along the path of the targeted OBU, and the targeted OBU has to move with the same velocity and in the same lane between any pair of adjacent receivers launched by the observer [1]. To protect the OBUs against this tracking attack, the DCS can efficiently be integrated with the random-encryption-period (REP) protocol proposed in [25]. In REPs, using group communications, an OBU surrounds itself with an encrypted communication zone to violate the conditions of being tracked by an observer.

4) Transparent roaming: Since any OBU can update its certificate from any RSU in the network, the DCS scheme overcomes the need to reregister the OBU entering a new domain with the new CA. Consequently, the transparent roaming is guaranteed in the DCS scheme.
5) Access control: Any illegal network access by a compromised RSU can be efficiently thwarted, since a CA can broadcast a revocation message including the certificates of that RSU. Upon receiving that revocation message, all the OBUs can disassociate themselves from that compromised RSU. Furthermore, all the OBU certificates issued by that RSU are revoked, as the revoked RSU certificates are contained in those certificates. In addition, a CA can revoke any misbehaving OBU by broadcasting a CRL containing the certificate of the misbehaving OBU. Consequently, all the network RSUs and OBUs terminate the communications with that OBU.

VIII. Performance Evaluation

In this section, we evaluate the performance of the DCS scheme according to different performance aspects.

A. OBU Certificate Update Delay

In this section, we compare the OBU certificate update delay in the DCS scheme, the ECPP protocol, and the classical PKI where an OBU has to contact a CA to update its certificates.

Let $T_{\text{cert-DCS}}, T_{\text{cert-ECPP}}$, and $T_{\text{cert-CA}}$ denote the time from the moment an OBU requests N_{cert} new certificates from an RSU until it receives the required certificates in the DCS scheme, the ECPP protocol, and the classical PKI, respectively. We consider the cryptography delay only due to the pairing and point multiplication operations on an elliptic curve, as they are the most time-consuming operations in the schemes under consideration. Let T_{pair} and T_{mul} denote the time required to perform a pairing operation and a point multiplication, respectively. In [26], T_{pair} and T_{mul} are found, for an MNT curve with embedding degree $k = 6$, to be equal to 4.5 and 0.6 ms, respectively. Let $T_{\text{crypt-DCS}}$ and $T_{\text{crypt-ECPP}}$ denote the total incurred cryptography delay from the moment an OBU requests N_{cert} new certificates from an RSU until it receives the required certificates in the DCS scheme and ECPP protocol, respectively. It should be noted that the cryptography delay (T_{crypt}) is part of the certificate update delay (T_{cert}) in any of the schemes under consideration. Table II gives the cryptography delay incurred in each step of the DCS certificate update algorithm, which is shown in Fig. 4, from the moment an OBU requests N_{cert} new certificates from an RSU, i.e., step 2, until it receives the required certificates, i.e., by the end of step 5. According to Table II, we have

TABLE II
DCS Certificate Update Cryptography Delay

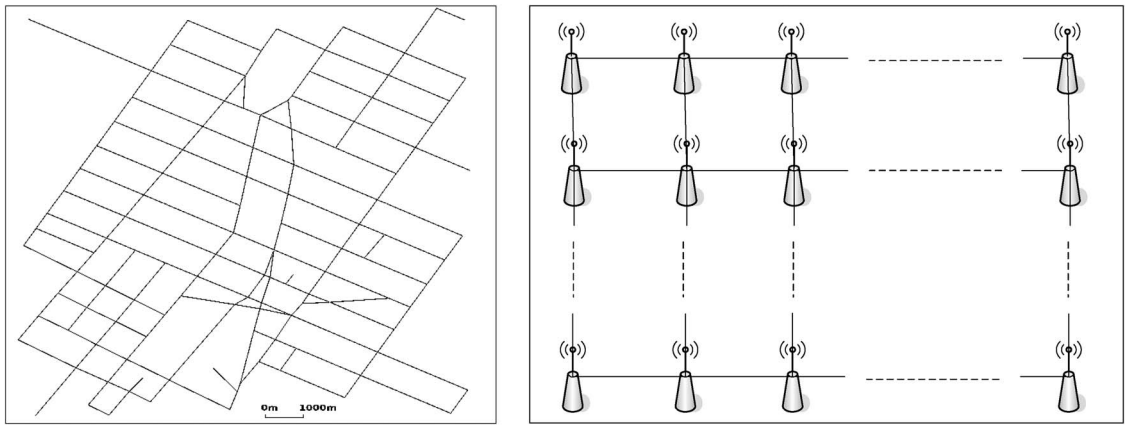
certificate update step	operation	entity involved	cryptography delay
step(2)	OBU_n certificate verification	RSU_j	$4T_{pair}+2T_{mul}$
	calculation of the shared key k_{jn}	RSU_j	T_{pair}
step(3)	calculation of N_{cert} partial public keys	RSU_j	$N_{cert}T_{mul}$
	calculation of N_{cert} partial secret keys	RSU_j	$N_{cert}T_{mul}$
step(4)	generation of N_{cert} final public keys	OBU_n	$N_{cert}T_{mul}$
step(5)	calculation of $\{U'_{n,r} \mid 1 \le r \le N_{cert}\}$	RSU_j	$N_{cert}T_{mul}$
	calculation of $\{L'_{n,r} \mid 1 \le r \le N_{cert}\}$	RSU_j	$N_{cert}T_{mul}$

$$T_{\text{crypt-DCS}} = 5T_{\text{pair}} + (2 + 5N_{\text{cert}})T_{\text{mul}}. \quad (13)$$

In the ECPP protocol [12], an RSU generates only one certificate for an OBU requesting certificate update. However, the ECPP protocol can easily be extended to enable an RSU to generate a bundle of N_{cert} certificates for the requesting OBU, which is similar to the DCS scheme. In the case where the ECPP protocol generates N_{cert} for the requesting OBU, we have

$$T_{\text{crypt-ECPP}} = (3 + 5N_{\text{cert}})T_{\text{pair}} + (4 + 9N_{\text{cert}})T_{\text{mul}}. \quad (14)$$

We have conducted two ns-2 [27] simulations to, respectively, compare the certificate update delay of the DCS scheme with that of the ECPP protocol and that of the classical PKI for the city street scenario shown in Fig. 5(a). The adopted simulation parameters are given in Table III, and the mobility traces are generated using TraNS [28]. We use the IEEE 802.11a standard, which is the basis of DSRC, to simulate the medium-access-control (MAC) protocol for VANETs [28], [29]. VANETs have two types of links: wireless links



(a) (b)

Fig. 5. Simulation scenario. (a) City street map. (b) RSU connection pattern.

TABLE III
NS-2 SIMULATION PARAMETERS

simulation area	$13.4\ Km \times 12.3\ Km$
simulation time	$100\ sec$
max. vehicle speed	$60\ km/h$
OBU transmission range	$300\ m$
MAC protocol	802.11a
OBU information dissemination interval	$300\ msec$
wired channel capacity	100 Mb/s
wireless channel capacity	6 Mb/s
number of RSUs	576
distribution of RSUs	uniform

connecting OBUs to each other and to the RSUs and wired links connecting the RSUs in one domain and the CA responsible for that domain, as shown in Fig. 3 (we consider only the domain of CA_i in Fig. 3). According to the DSRC specifications, each wireless data channel in a VANET has a bandwidth of 10 MHz corresponding to a channel data rate in the range of 3–27 Mb/s [30]. We select a data rate of 6 Mb/s for the wireless channels in VANETs. We consider the links of the Ethernet connecting the RSUs and CA_i to have a data rate of 100 Mb/s. The RSU connection pattern employed in our simulation is shown in Fig. 5(b). The adopted RSU connection considers a well-deployed VANET, where the RSUs are uniformly distributed, with the distance between any pair of adjacent RSUs being 500 m. CA_i is located at the top-left corner of the city scenario shown in Fig. 5(a). To simulate real-life VANET scenarios, we conduct the certificate update scenarios imposed on VANET safety-related applications, where each OBU has to disseminate information about the road condition every 300 ms, according to DSRC.

The first simulation is conducted to compare the certificate update delay in the DCS scheme with that in the ECPP protocol. Fig. 6(a) shows the certificate update delay in milliseconds for the DCS scheme and the ECPP protocol versus the simulation time in seconds. In the conducted simulation, we consider N_{cert} to be constant for all the OBUs, where we consider values of N_{cert} equal to one, five, and ten certificates. In addition, an OBU sends a certificate update request every 10 s during the simulation, and the corresponding certificate update delay is measured. The variations in $T_{\text{cert-DCS}}$ and $T_{\text{cert-ECPP}}$ are due to the variations in the distance separating the OBU requesting the certificate update and the RSU issuing the certificate. Table IV shows the average values of the certificate update delay shown in Fig. 6(a). It can be seen from Table IV that the DCS scheme outperforms the ECPP protocol and the percentage of the delay savings obtained by DCS compared with the ECPP increases with N_{cert}. It should be noted that the average values for $T_{\text{cert-DCS}}$ and $T_{\text{cert-ECPP}}$ in Table IV are independent on the density of the RSUs, as only one RSU is involved in each certificate update process. Therefore, the RSU density has no effect on the certificate-update delay.

The second simulation is conducted to compare the certificate update delay of the DCS scheme with that of the classical PKI [8] under a well-deployed VANET. The classical PKI certificate update requires each OBU requesting certificate update to contact the CA through the RSUs, as the CA is the only entity responsible for generating the certificates. The elliptic-curve digital signature algorithm (ECDSA) [31] is the classical PKI digital signature method chosen by the WAVE standard, where a certificate and signature verification takes $4T_{\mathrm{mul}}$, and a signature generation takes T_{mul}.

We consider two certificate update scenarios, shown in Fig. 3, as follows. The first scenario is the classical PKI certificate update, where OBU_m (shown in red) needs to update its certificates. Hence, it should send a certificate update request to CA_i via the nearest RSU, which in this case is RSU_1. After the request reaches RSU_1, it will be forwarded through the RSUs' Ethernet to CA_i, where the request message experiences a delay of $4T_{\mathrm{mul}}$ at each intermediate RSU, as each RSU has to verify the certificate and the signature of the sender before forwarding the request; otherwise, a denial-of-service attack can easily be launched by sending faked requests, which can overwhelm CA_i. When the certificate update request reaches CA_i, it has to verify the request, which takes $4T_{\mathrm{mul}}$, and generate new N_{cert} certificates for OBU_m, which takes $N_{\mathrm{cert}}T_{\mathrm{mul}}$. Then, CA_i forwards the new certificates to RSU_l, which, in turn, forwards them to OBU_m. In the second scenario, OBU_m updates its certificates directly from RSU_1, as proposed by the DCS scheme.

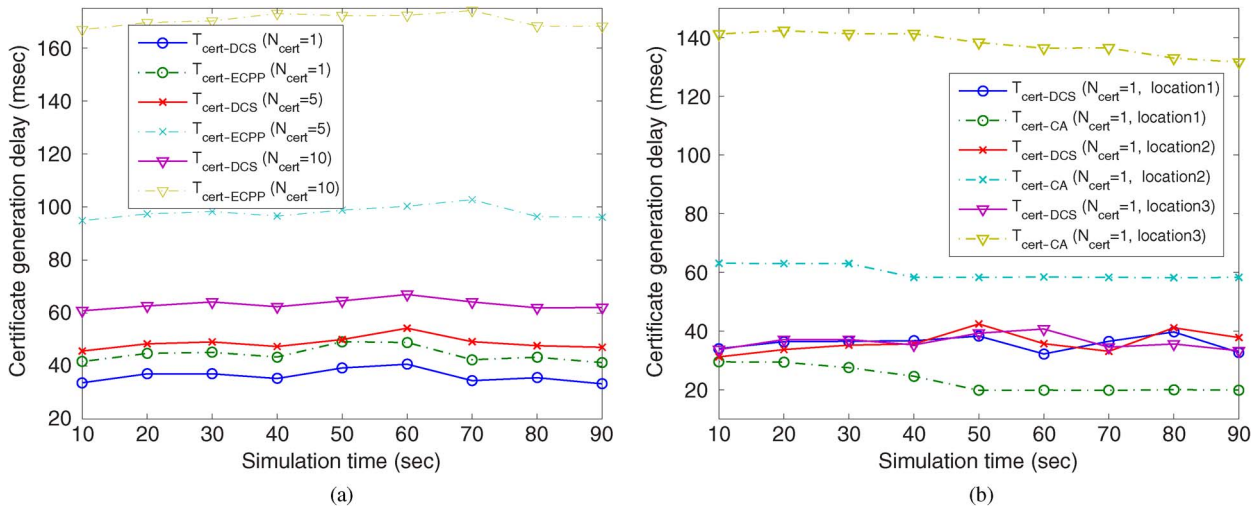


Fig. 6. Certificate-update delay. (a) Certificate-update delay for DCS and ECPP. (b) Certificate-update delay for DCS and classical PKI.

TABLE IV
AVERAGE CERTIFICATE-UPDATE DELAY

N_{cert}	avg. $T_{cert-DCS}$ (msec)	avg. $T_{cert-ECPP}$ (msec)	delay-saving
1	36.2	44.4	18.5%
5	48.8	98	50.2%
10	63.3	170.5	62.9%

Fig. 6(b) shows the classical PKI certificate-update delay $T_{\text{cert-CA}}$ and the DCS certificate-update delay $T_{\text{cert-DCS}}$ in milliseconds versus the simulation time. We conducted simulation for the two certificate update scenarios triggered by OBU_m for N_{cert} equal to 1 at three different locations (location1, location2, and location3), corresponding to initial distances of 2.7, 4.7, and 10.3 km, respectively, from CA_i at the beginning of the simulation. The certificate update process is triggered every 10 s during the simulation, and the corresponding certificate update delay is measured. The variations in $T_{\text{cert-CA}}$ are due to the number of the intermediate RSUs existing in the connection between CA_i and OBU_m. It can be seen that $T_{\text{cert-DCS}}$ is almost the same for the three locations and is confined within the range 31–43 ms. This is due to the fact that the DCS scheme is independent on CA_i. On the other hand, it can be seen that $T_{\text{cert-CA}}$ increases with the distance from CA_i. Consequently, the delay savings of the proposed DCS scheme compared with the classical PKI certificate-update increase with the distance from the CA. For example, the average certificate update delay is 59.87 ms for location2, while that for the DCS scheme is 36.2 ms. Consequently, the DCS scheme decreases the certificate-update delay by 39.54% compared with the classical PKI in that case. From the aforesaid discussion, it can be seen that, even under a well-deployed VANET, the DCS scheme outperforms the classical PKI in terms of certificate-update delay, which directly translates into a better certification service. In addition, since in the classical PKI, all certificate updates are handled by the CA, it is expected that the certificate-update delay from the CA increases in real-life large-scale VANETs.

B. Successful Certification Ratio

When an OBU_m requests N_{cert} certificates from an RSU_l, RSU_l should process the request, generate the required certificates, and deliver them to OBU_m before OBU_m moves out of the communication range of RSU_l; otherwise, the certificate-update process fails. Therefore, if the number of certificate-update requests is large, the RSU will not be able to process all the requests, and some requests may be dropped. To calculate the maximum number of certificates that an RSU can generate within its coverage range, we adopt the following formula [12]:

$$NC_{\max} = \frac{R}{\overline{S} \cdot T_{\text{cert}}} \tag{15}$$

where $NC_{\max}$ is the maximum number of certificates an RSU can generate within its coverage range R, $\overline{S}$ is the average speed of the OBUs within R, and T_{cert} is the average certificate-update delay of the scheme under consideration.

The successful certification ratio (SCR) is the metric usually used to evaluate the efficiency of authentication algorithms [32]. The SCR is defined as the ratio of the number of successful certificate generations (NC_s) to the number of total certificate requests (NC_t). Hence, we have

$$\text{SCR} = \begin{cases} 1, & \text{if } NC_s \le NC_{\max} \\ \frac{NC_s}{NC_t}, & \text{if } NC_s > NC_{\max}. \end{cases} \tag{16}$$

We consider an RSU with $R = 600$ m (corresponding to an omnidirectional communication range with radius 300 m according to DSRC), and the average speed of OBUs is $\overline{S} =$ 60 km/h. Fig. 7 shows the successful certification ratio for the DCS scheme and the ECPP protocol [12] for values of N_{cert} equal to one, five, and ten certificates versus the total number of certificate requests, where we used the values of T_{cert} in Table IV. It should be noted that, in the cases where $N_{\text{cert}} > 1$, each request in Fig. 7 corresponds to generating N_{cert} certificates. It can be seen that DCS gives a higher SCR than the ECPP protocol. Furthermore, the SCR for DCS with $N_{\text{cert}} =$ 10 is even higher than that of the ECPP with $N_{\text{cert}} = 5$.

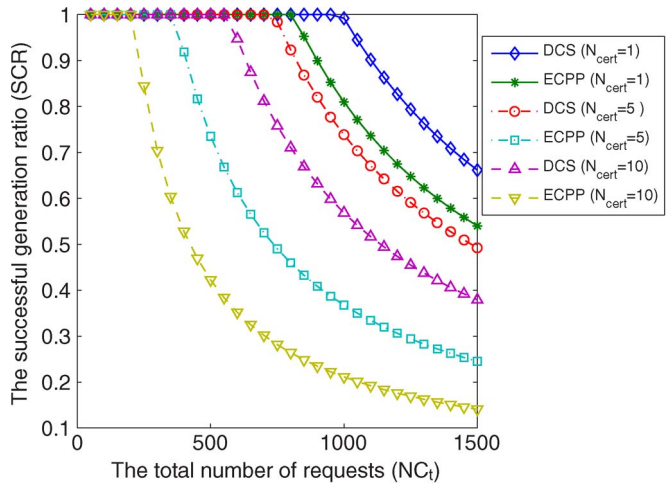


Fig. 7. Successful certification ratio.

Since DCS can handle a larger number of certificate requests than ECPP for the same duration, the DCS is more suitable for the requirement of vehicular networks.

C. Required RSU Density in DCS

In this section, we give an estimate of the required RSU density in the DCS scheme. It is more meaningful to express the RSU density $(density_{\mathrm{RSU}})$ as the number of RSUs per road unit length (in kilometers) instead of per unit area (in square kilometers) as RSUs are implemented only on the roads, and a road width is generally much smaller than its length. The average distance D_{RSU} the OBUs can move with no need to contact an RSU is

$$D_{\mathrm{RSU}} = \frac{1}{60}\overline{N}_{\mathrm{cert}}\overline{vperiod}\,\overline{S}\ (\mathrm{km}) \tag{17}$$

where $\overline{N}_{\mathrm{cert}}$ is the average number of the generated certificates per OBU from the RSUs, $\overline{vperiod}$ is the average validity period of the OBU certificates in minutes, and $\overline{S}$ is the average speed of the OBUs in kilometers per hour. It should be noted that the parameters in (17) correspond to only one CA domain. Since D_{RSU} can be interpreted as the road distance between two adjacent RSUs. Consequently, the required RSU density $(density_{\mathrm{RSU}})$ for the DCS scheme can be calculated as

$$density_{\mathrm{RSU}} = \frac{1}{D_{\mathrm{RSU}}} = \frac{60}{\overline{N}_{\mathrm{cert}}\overline{vperiod}\,\overline{S}}\ (/\mathrm{km}). \tag{18}$$

Equation (18) can be used in the design phase of the DCS scheme to calculate the number of RSUs needed for the operation of the DCS scheme.

Table V gives an example of the required densities and numbers of RSUs for the states of New York and Hawaii with $\overline{vperiod} = 1$ min and $\overline{S} = 60$ km/h. New York has an area of 141 299 km^2, while that for Hawaii is 28 311 km^2 [33]. The total length of the urban and rural roads is obtained from [34]. Since the density of the OBUs in an urban road is higher than that in a rural road, it will not be cost-effective to implement RSUs in rural roads with a density equal to that in urban roads. Therefore, we select the $\overline{N}_{\mathrm{cert}}$ for rural and urban roads to be

TABLE V
EXAMPLE OF THE REQUIRED $density_{\mathrm{RSU}}$ IN DCS WITH $\overline{vperiod} = 1$ min AND $\overline{S} = 60$ km/h

state	New York	Hawaii
rural roads length (Km)	106014	3285
urban roads length (Km)	77033	3701
$\overline{N}_{cert}$ (rural)	20	20
$\overline{N}_{cert}$ (urban)	10	10
$density_{RSU}$ (rural)	0.05	0.05
$density_{RSU}$ (urban)	0.1	0.1
number of required RSUs (rural)	5301	165
number of required RSUs (urban)	7074	371
total number of RSUs	13005	536

TABLE VI
RSU_j CERTIFICATE SIZE IN DCS

parameter	PK_{j_i}	U_j	V_j	PID_j	Q_i	$cert_{RSU_{ji}}$
size in bytes	21	21	21	8	21	92

TABLE VII
OBU_m CERTIFICATE SIZE IN DCS

parameter	size in bytes
$PK_{m_{ji}}$	21
$U_m^{\backprime}$	21
$V_m^{\backprime}$	21
$vperiod$	4
PID_m	8
$cert_{RSU_{ji}}$	92
$cert_{OBU_{mji}}$	167

20 and 10, respectively. The total number of the required RSUs can be decreased by increasing the validity period $(\overline{vperiod})$ of the certificates of the OBUs or increasing N_{cert}. However, increasing $\overline{vperiod}$ increases the probability of being tracked, i.e., lowering the privacy protection level. Furthermore, increasing the number of certificates (N_{cert}) generated from RSUs decreases the SCR, as shown in Fig. 7. A compromise between the privacy protection level and the SCR of RSUs should be made according to the required RSU density. It should be noted that each CA can change the minimum and maximum bounds to the value of the certificate-validity period according to the required level of privacy protection and broadcast these bounds to the RSUs in its domain through its local Ethernet.

D. Communication Overhead

We consider the Tate pairing implementation on an MNT curve with embedding degree 6, where $\mathbb{G}_1$ is represented by 161 bits. Accordingly, each point on this MNT curve is represented by 21 B. Tables VI and VII give each parameter and the corresponding size in bytes for an RSU and OBU certificate, respectively. The last column and row in Tables VI and VII gives the total size of the certificate under consideration, respectively. It can be seen that an RSU has a certificate size of 92 B, while that for an OBU is 167 B.

It is indicated in Section VI-A that an OBU_m with $cert_{\mathrm{OBU}_{mji}}$ can generate a valid signature $(U^{\backprime}, V^{\backprime})$ for an arbitrary message M. Since $U^{\backprime}$ and $V^{\backprime}$ are points on the elliptic curve, the signature size in DCS is 42 B. Consequently, the communication overhead incurred in a signed message transmitted by an OBU is 209 B, which is the certificate size

TABLE VIII
SIGNING AND VERIFICATION DELAY

method	message signing	one signature and certificate verification	K signatures and certificate verification
ECDSA	T_{mul}	$4T_{mul}$	$4KT_{mul}$
BLS	$T_{mul} + T_{mtp}$	$4T_{pair} + 2T_{mtp}$	$(2K+2)T_{pair} + 2KT_{mtp}$
CAS	$2T_{mul} + T_{mtp}$	$5T_{pair} + 2T_{mtp}$	$(4K+1)T_{pair} + 2KT_{mtp}$
ECPP	T_{mul}	$3T_{pair} + 11T_{mul}$	$3KT_{pair} + 11KT_{mul}$
DCS	$2T_{mul}$	$5T_{pair} + 3T_{mul}$	$5T_{pair} + 3KT_{mul}$

plus the signature size, compared with an overhead of 189 B in the ECPP protocol. According to the WAVE standard [8], the maximum payload data size in a signed message is 65.6 kB. Consequently, the ratio of the communication overhead incurred by the DCS scheme to the payload data size is 0.3%, which means that the DCS scheme is feasible with respect to the incurred communication overhead.

E. OBU Message Signing Delay

In DCS, the signature of an OBU_m with $cert_{\mathrm{OBU}_{mji}}$ on an arbitrary message M is (U'', V''). The cryptography operation involved in calculating either U'' or V'' is point multiplication. Therefore, the total delay for signing a message in DCS is $2T_{\mathrm{mul}}$. The second column in Table VIII gives the message-signing delay for ECDSA, BLS, CAS, ECPP, and DCS. BLS is a pairing-based aggregate signature [35]. CAS is a certificateless aggregate signature scheme [36], which is the basis of the DCS batch-verification scheme.

It can be seen that ECDSA and ECPP give the lowest message-signing delay, and DCS gives the second lowest delay. The effect of the message-signing delay is alleviated by the fact that an OBU has to disseminate only one signed message every 300 ms, which means that an OBU has a time window of 300 ms to prepare a signature on a message. The DCS scheme has a message-signing delay of 1.2 ms, which can be neglected, compared with the time window an OBU has to sign a message.

F. Batch-Verification Delay

We compare the verification delay of the DCS batch signature and certificate-verification scheme with ECDSA, BLS, CAS, and ECPP.

The time needed to verify one ECDSA signature is $2T_{\mathrm{mul}}$, and that for BLS is $2T_{\mathrm{pair}} + T_{\mathrm{mtp}}$, where T_{mtp} is a map-to-point hash function. T_{mtp} is found, for an MNT curve, to be 3.9 ms [37]. We consider the verification delay for a certificate sent with a message signature for ECDSA and BLS to be equal to that of a signature verification. The time needed to verify one CAS signature is $3T_{\mathrm{pair}} + 2T_{\mathrm{mtp}}$. For CAS, there is no certificate; however, to verify the sender, a check process must be performed, which takes $2T_{\mathrm{pair}}$. For ECPP, the total verification delay of a certificate and signature is $3T_{\mathrm{pair}} + 11T_{\mathrm{mul}}$. For the DCS scheme, the verification delay of a certificate and message signature requires $5T_{\mathrm{pair}} + 3T_{\mathrm{mul}}$, where $5T_{\mathrm{pair}}$ corresponds to the pairing operations in the left- and right-hand sides of (12), and $3T_{\mathrm{mul}}$ corresponds to the point multiplication operations in $\overline{U}, \overline{U}'$, and $\overline{U}''$. Table VIII shows a summary of the verification delays for the ECDSA, BLS, CAS, ECPP, and DCS schemes.

Fig. 8(a) shows the verification delay in milliseconds versus the number of the received messages. It can be seen that the DCS scheme has the lowest verification delay. Furthermore, from Table VIII and the values of T_{pair}, T_{mtp}, and T_{mul}, the most time-consuming operation in the signature-verification process of the schemes under consideration is the pairing operation. Hence, the reason for the superiority of the DCS is that the number of the pairing operations required for signatures verification is independent of the number of the signatures to be verified. The maximum number of signatures and certificates that can simultaneously be verified in 300 ms is 11, 14, 17, 124, and 154 messages for the CAS, ECPP, BLS, ECDSA, and DCS schemes, respectively. The number of signatures and certificates that the DCS scheme can verify is greater than that of ECDSA by 24.2%. Fig. 8(b) shows the delay for batch signature verification, batch certificate verification, and simultaneous batch signature and certificate verification. The maximum number of certificates that can aggregately be verified within 300 ms is 234 certificates, while that for signatures is 477 signatures.

To further evaluate the DCS batch verification scheme, we conduct ns-2 [27] simulation using the same parameters in Table III, except for simulation area and time, which become 7.4 km $\times$ 7.4 km and 30 s, respectively. In this simulation, we are interested in the message loss incurred by OBUs due to V2V communications only, i.e., we do not consider the implementation of RSUs. The average message loss ratio is defined as the average ratio between the number of messages dropped every 300 ms, due to signature- and certificate-verification delay, and the total number of messages received every 300 ms. According to DSRC, each OBU has to disseminate information about the road condition every 300 ms. To properly and instantly react to the varying road conditions, each OBU should verify the messages received during the last 300 ms before disseminating a new message about the road condition. Therefore, we chose to measure the message loss ratio every 300 ms. Fig. 9 shows the analytical and simulated average message loss ratio versus the average number of OBUs within the communication range of each OBU for DCS, ECPP, ECDSA, BLS, and CAS, respectively. It can be seen that the simulated average message loss ratio closely follows the analytical message loss ratio, which is calculated based on the maximum number of messages that can be verified within 300 ms in the schemes under consideration. The difference between the analytical and simulations results stems from observing that some zones in the simulated area become more congested than other zones; thus, some OBUs experience a higher message loss than other OBUs, which, on average, leads to that difference between the analytical and simulation results. Furthermore, the proposed DCS batch verification provides the lowest message loss ratio, and the message loss ratio increases as the number of OBUs within communication range increases. The reason for the superiority of the DCS scheme is that it can aggregately verify a number of signatures higher than that of ECPP, ECDSA, BLS, or CAS.

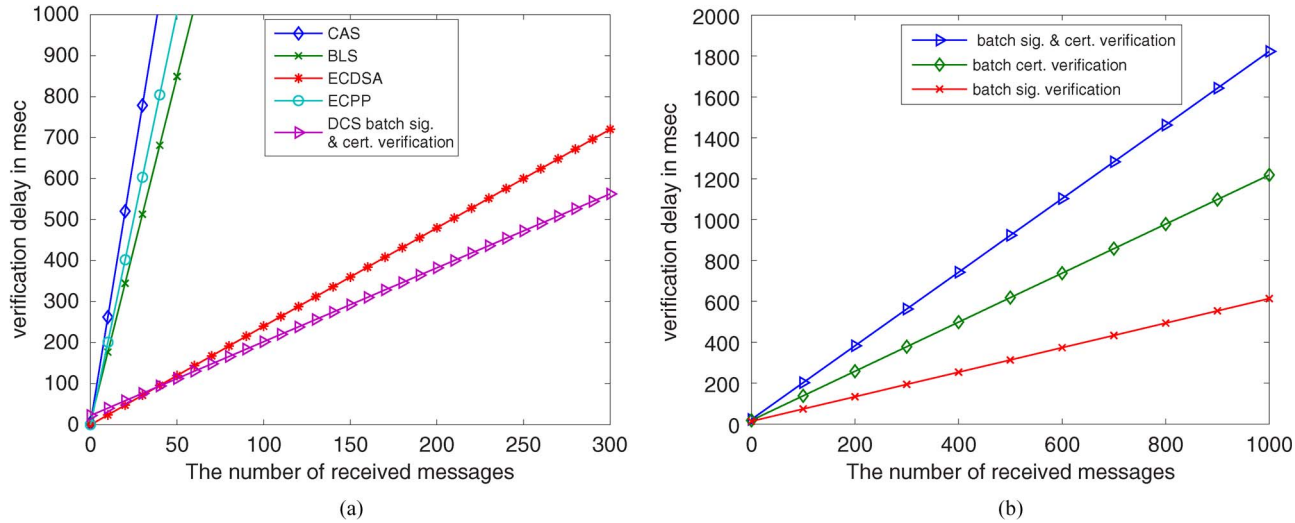


Fig. 8. Verification delay. (a) Verification delay comparison between different schemes. (b) Verification delay comparison between different batch schemes of DCS.

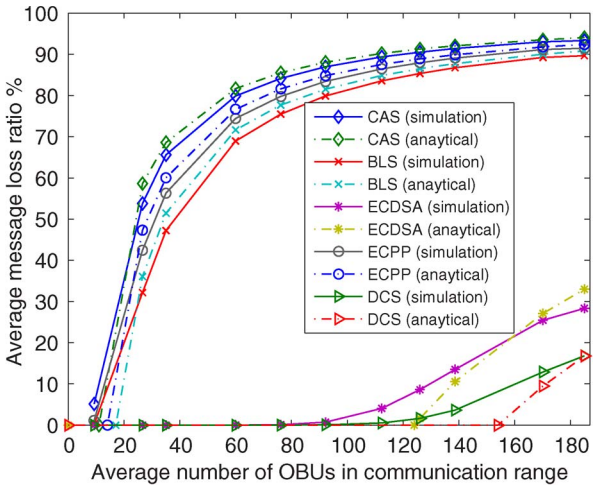


Fig. 9. Comparison between message loss ratios for different schemes.

G. Additional GPS Memory Requirements

In the DCS scheme, the GPS receiver in each OBU is required to be loaded with the geographic coordinates of the RSUs, which incurs additional memory requirements. According to [8], each latitude or longitude coordinate of the geographic location of an RSU is represented by 4 B. With the results obtained in Section VIII-C, the number of RSUs in a CA domain is on the order of 10^4. Consequently, the memory size required to save the coordinates of the RSUs in a domain is 0.08 MB. Most of the currently available GPS receivers have memory storage sufficient to meet this requirement.

IX. Conclusion

In this paper, we have proposed an efficient DCS scheme for vehicular communications, which offers flexible interoperability to avoid the key escrow issue in different administrative authorities and an efficient distributed algorithm for any OBUs to update or revoke its certificate from the available RSUs in a timely manner. In addition, with the batch verification, the entities in the DCS scheme can rapidly simultaneously verify a large number of message signatures and certificates. Therefore, the proposed DCS scheme can significantly reduce the complexity of certificate management and achieve excellent efficiency and scalability, particularly when it is deployed in heterogeneous vehicular networks. For our future work, we will investigate the revocation issue under the context of the proposed DCS scheme.

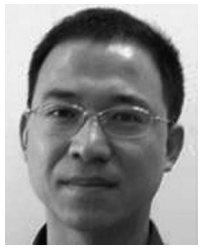
References

[1] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Security*, vol. 15, no. 1, pp. 39–68, Jan. 2007.
[2] F. Dötzer, "Privacy issues in vehicular ad hoc networks," in *Proc. 2nd ACM Workshop Veh. Ad Hoc Netw.*, Sep. 2006, pp. 197–209.
[3] P. Papadimitratos, A. Kung, J. P. Hubaux, and F. Kargl, "Privacy and identity management for vehicular communication systems: A position paper," in *Proc. Workshop Standards Privacy User-Centric Identity Manage.*, Zurich, Switzerland, Jul. 2006.
[4] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "CARAVAN: Providing location privacy for VANET," in *Proc. ESCAR*, Nov. 2005.
[5] J. P. Hubaux, "The security and privacy of smart vehicles," *IEEE Security Privacy*, vol. 2, no. 3, pp. 49–55, May/Jun. 2004.
[6] M. Shi, X. Shen, and J. Mark, "IEEE 802.11 roaming and authentication in wireless LAN/cellular mobile networks," *Wireless Commun.*, vol. 11, no. 4, pp. 66–75, Aug. 2004.
[7] Y. Jiang, C. Lin, X. Shen, and M. Shi, "Mutual authentication and key exchange protocols for roaming services in wireless mobile networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 9, pp. 2569–2577, Sep. 2006.
[8] *IEEE Trial-Use Standard for Wireless Access In Vehicular Environments—Security Services for Applications and Management Messages*, IEEE Std. 1609.2-2006, 2006.
[9] *5.9 GHz DSRC*. [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/index.html
[10] 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.
[11] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Crypto*, 2004, vol. 3152, pp. 41–55.
[12] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. INFOCOM*, 2008, pp. 1229–1237.
[13] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: A robust signature scheme for vehicular networks using binary authentication tree," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1974–1983, Apr. 2009.
[14] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. 21st Annu. Int. Cryptology Conf. Adv. Cryptol.*, 2001, pp. 213–229.

[15] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, Sep. 2004.
[16] M. Scott, "Computing the Tate pairing," in *Topics in Cryptology*. Berlin, Germany: Springer-Verlag, 2005, pp. 293–304.
[17] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reductions," *IEIC Tech. Rep.*, vol. 100, no. 323, pp. 99–108, 2000.
[18] *Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL)*. [Online]. Available: http://www.shamus.ie/
[19] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Des. Codes Cryptogr.*, vol. 19, no. 2, pp. 173–193, Mar. 2000.
[20] D. Boneh and R. Lipton, "Algorithms for black-box fields and their application to cryptography," in *Proc. Adv. Cryptol.*, 1996, pp. 283–297.
[21] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," in *Proc. Adv. Cryptol.*, 2003, pp. 452–473.
[22] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from Asiacrypt 2003," in *Proc. 4th Int. Conf. CANS*, Berlin, Germany: Springer-Verlag, 2005, vol. 3810, pp. 13–25.
[23] K. P. Laberteaux, J. J. Haas, and Y. Hu, "Security certificate revocation list distribution for VANET," in *Proc. 5th ACM Int. Workshop Veh. Inter-NETw.*, 2008, pp. 88–89.
[24] R. Oppliger, "Protecting key exchange and management protocols against resource clogging attacks," in *Proc. Joint Working Conf. Secure Inf. Netw.*, 1999, pp. 163–175.
[25] A. Wasef and X. Shen, "REP: Location privacy for VANETs using random encryption periods," *Mobile Netw. Appl. (MONET)*. DOI: 10.1007/s11036-009-0175-4, to be published.
[26] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, 2008, 246–250.
[27] *The network simulator–ns-2*. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/User Information
[28] *Traffic and network simulation environment–TraNS*. [Online]. Available: http://trans.epfl.ch/
[29] D. Cottingham, I. Wassell, and R. Harle"Performance of IEEE 802.11a in vehicular contexts," in *Proc. IEEE 65th Veh. Technol. Conf.*, 2007, pp. 854–858.
[30] J. Yin, T. El Batt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proc. 1st ACM Int. Workshop Veh. Ad Hoc Netw.*, 2004, pp. 1–9.
[31] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Security*, vol. 1, no. 1, pp. 36–63, 2001.
[32] K. Sadasivam and T. Yang, "Evaluation of certificate-based authentication in mobile ad hoc networks," in *Proc. IASTED NCS*, Apr. 2005.
[33] *List of U.S. states by area*. [Online]. Available: http://www.knowledgerush.com/kr/encyclopedia/List_of_U.S._states_by_area/
[34] *United States Department of Transportation—Federal Highway Administration*. [Online]. Available: http://www.fhwa.dot.gov/policyinformation/statistics/2007/hm20m.cfm
[35] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. Adv. Cryptology—EUROCRYPT*, 2003, pp. 416–432.
[36] Z. Gong, Y. Long, X. Hong, and K. Chen, "Two certificateless aggregate signatures from bilinear maps," in *Proc. 8th ACIS Int. Conf. SNPD*, 2007, vol. 3, pp. 188–193.
[37] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, "A novel anonymous mutual authentication protocol with provable link-layer location privacy," *IEEE Trans. Veh. Technol.*, vol. 58, no. 3, pp. 1454–1466, Mar. 2009.

Albert Wasef (M'09) received the B.Sc. and M.Sc. degrees in electrical communications engineering from El Menoufia University, El Menoufia, Egypt, in 1998 and 2003, respectively. He is currently working toward the Ph.D. degree with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

He is also currently a member of the Broadband Communications Research Group, University of Waterloo. His research interest includes wireless network security, privacy preservation in vehicular networks, and group communications.



Yixin Jiang received the Ph.D. degree from the Department of Computer Science, Tsinghua University, Beijing, China, in 2006.

He is currently an Associate Professor with Tsinghua University. During 2007–2009, he was a Postdoctoral Fellow with the University of Waterloo, Waterloo, ON, Canada. In 2005, he was a Visiting Scholar with the Department of Computer Sciences, Hong Kong Baptist University, Hong Kong. His current research interests include wireless network security, trusted computing, and network coding.

Prof. Jiang has served as a Technical Program Committee Member for network conferences, such as the IEEE International Conference on Computer Communications and Networks, the IEEE Global Communications Conference, the IEEE International Conference on Communications, and the IEEE Wireless Communications and Networking Conference.

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

He is currently a Professor and the University Research Chair with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is the author or a coauthor of three books and more than 400 papers and book chapters on wireless communications and networks, control, and filtering. He serves as the Editor-in-Chief for *Peer-to-Peer Networking and Application* and an Associate Editor for *Computer Networks*, *ACM/Wireless Networks*, and *Wireless Communications and Mobile Computing*. He has also served as a Guest Editor for *ACM Mobile Networks and Applications*. His research focuses on mobility and resource management in interconnected wireless/wired networks, ultra-wideband wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks.

Dr. Shen is a Registered Professional Engineer in the Province of Ontario, and a Distinguished Lecturer of IEEE Communications Society. He received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award in 2003 from the Province of Ontario, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He served as the Tutorial Chair for the 2008 IEEE International Conference on Communications, the Technical Program Committee Chair for the 2007 IEEE Global Telecommunications Conference, the General Cochair for the 2007 International Conference in Communications and Networking in China and the 2006 International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, and the Founding Chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the KICS/IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He has also served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and the *IEEE Communications Magazine*.