EPA: an Efficient and Privacy-Aware revocation Mechanism for Vehicular Ad Hoc Networks

Carlos Gañán[a], Jose L. Muñoz[a], Oscar Esparza[a], Jorge Mata[a], Juanjo Alins[a]

[a] *Universitat Politècnica de Catalunya Departament Enginyeria Telemàtica 1-3 Jordi Girona, C3 08034 Barcelona (Spain)*

Abstract

Security is vital for the reliable operation of vehicular ad hoc networks (VANETs). One of the critical security issues is the revocation of misbehaving vehicles. While essential, revocation checking can leak private information. In particular, repositories receiving the certificate status queries could infer the identity of the vehicles posing the query and the target of the query. An important loss of privacy results from this ability to tie the checking vehicle with the query's target, due to their likely willingness to communicate. In this paper, we propose an Efficient and Privacy-Aware revocation Mechanism (EPA) based on the use of Merkle Hash Trees (MHT) and a Crowds-based anonymous protocol, which replaces the time-consuming certificate revocation lists checking process. EPA provides explicit, concise, authenticated and unforgeable information about the revocation status of each certificate while preserving the users' privacy. Moreover, EPA reduces the security overhead for certificate status checking, and enhances the availability and usability of the revocation data. By conducting detailed performance evaluation, EPA is demonstrated to be reliable, efficient, and scalable.

Keywords: Revocation, Privacy, Merkle Hash Tree, Crowds, VANET.

1. Introduction

Vehicular ad hoc networks (VANETs) aim at enhancing safety and efficiency in transportation systems. VANETs consist of entities including On-Board Units (OBUs) and infrastructure Road-Side Units (RSUs). Mobile nodes are capable of communicating with each other (i.e., Vehicle to Vehicle Communication -V2V communication) and with the RSUs (i.e., Vehicle to Infrastructure Communication -V2I communication). As any other wireless network, VANETs can be vulnerable to attacks and jeopardize users' privacy. For instance, an attacker could inject false information, or collect vehicles' messages, track their locations, and infer sensitive user data. To thwart such attacks, security and privacy enhancing mechanisms are necessary or, in fact, a prerequisite for deployment. According to the IEEE 1609.2 standard [1], vehicular networks will rely on the public key infrastructure (PKI). In PKI, a certification authority issues an authentic digital certificate for each node in the network. Due to misbehavior, intentional or otherwise, certificates need to be revoked in order to limit the risk that potential misuse poses to the rest of the network. The IEEE 1609.2 standard [1] states that VANETs will depend on certificate revocation lists (CRLs) to achieve revocation. CRLs are black lists that enumerate revoked certificates along with the date of revocation and, optionally, the reasons for revocation.

Unfortunately, the CRL size in VANETs is expected to be large for the following reasons: (i) the scale of VANET will be significantly large. (ii) to preserve the privacy of the drivers (i.e., to prevent the leakage of the real identities and location information of the drivers from any external eavesdropper) each OBU should be preloaded with a set of anonymous digital certificates, where the OBU has to periodically change its anonymous certificate to mislead attackers. Consequently, a revocation of an OBU results in revoking all the certificates carried by that OBU leading to a large increase in the CRL size [2, 3, 4]. Thus, distributing and updating CRLs raise a challenge. Several CRLs distribution protocols have been proposed to palliate this pitfall, e.g., using compressed CRLs by using Bloom filters [5]. Other proposals suggest the use of regional CAs and short lived certificates to decrease the number of entries in the CRL [6]. However, these works overlooked the disruption nature of vehicular networks. Recently, some work have appeared dealing with the distribution of certificate status information (CSI) in environments prone to disruption [7, 8, 9, 10]. These mechanisms take advantage of caching strategies combined with hashing techniques to enhance the availability of the revocation service. Nevertheless, none of these approaches takes into account the loss of privacy due to the CSI checking process.

Regardless of their particulars, current revocation meth-

Email addresses: carlos.ganan@entel.upc.edu (Carlos Gañán), jose.munoz@entel.upc.edu (Jose L. Muñoz), oesparza@entel.upc.edu (Oscar Esparza), jorge.mata@entel.upc.edu (Jorge Mata), juanjo@entel.upc.edu (Juanjo Alins)

ods that differ from the traditional CRL approach have an unpleasant side-effect: they divulge too much information [11]. In particular, a non-trusted third party (e.g., a RSU) could gain knowledge about who is talking to whom, by just analyzing the CSI requests. This is significant, because the revocation status check typically serves as a prelude to actual communication between the two parties. Hence, RSUs could acquire significant statistics of the PKI such as who sends a message to whom, how often, etc. Recently, there have appeared some works that intend to provide privacy during the revocation process [12, 13]. However, they mainly use CRLs to convey the revocation information. Though CRLs prevent the user's privacy, they consume too much bandwidth when transmitted.

In this article, we address the issue of checking the status of a certificate by exploiting the use of Merkle hash trees (MHT) [14]. MHT have already been suggested as means to provide an efficient revocation service ([9, 15]). However, they were used in such a way that each time a certificate had to be checked, users had to contact a local repository to verify its validity. We propose an Efficient and Privacy-Aware (EPA) revocation mechanism that uses MHT to provide certificate no-invalidity proofs (i.e., a proof that a given certificate is not revoked) that each vehicle stores locally. Thus, EPA allows vehicles proving the no-invalidity of their certificates to other entities. CAs will transmit an extended CRL to the RSUs that will act as repositories. RSUs will construct a MHT from the information contained in the CRL. Then, any vehicle will be able download the corresponding certificates' no-invalidity proofs. For enforcing anonymity in multi-hop VANETs, vehicles using EPA do not contact directly the RSU when updating the CSI. In contrast, they follow a Crowds-like protocol [16] according to which each user probabilistically decides to send a message directly to a common receiver, or else to forward it to a peer, who is asked to repeat the process. Our protocol differs from the original Crowds in that, first, it does take into account transmission losses, and secondly, it is specifically conceived for multi-hop VANETs, rather than for wired networks.

EPA enjoys three main advantages over the traditional revocation mechanisms: i) EPA saves dramatically on bit transmissions and costs, i.e., vehicles do not have to download the whole CRL, just positive proofs of their certificates' no-invalidity; ii) EPA always provides a positive statement about the no-invalidity status of each not-yet-expired certificate; iii) EPA always allows a complete answer to any possible query of a user to the RSU and without trusting the latter in any special way. Thus, EPA decreases the dependency on the infrastructure to provide the certificate status checking service at the same time that prevents RSUs to acquire any private information. Once the vehicles have obtained these short proofs asserting the no-invalidity of their certificates, they do not need to contact the infrastructure anymore. To obtain and update these proofs, vehicles contact RSUs without leaking personal information. Therefore, EPA provides explicit, concise, authenticated and unforgeable information about the revocation status of each certificate while preserving the users' privacy.

The rest of this article is organized as follows. In Section 2 we summarize the related work regarding CSI management. Section 3 describes the Efficient and Privacy-Aware (EPA) revocation mechanism. In Section 4 we evaluate and compare our proposal to other revocation mechanisms. Finally, we conclude in Section 5.

2. Background

In this section, we describe existing revocation proposals for VANET.

2.1. Privacy aware revocation approaches for VANET

The IEEE 1609.2 standard [1] proposes an architecture based on the existence of a Trusted Third Party (TTP), which manages the revocation service. In this architecture each vehicle possesses several short-lived certificates (used as pseudonyms), to ensure users' privacy. However, short-lived certificates are not enough as compromised or faulty vehicles could still endanger other vehicles until the end of their certificate lifetimes. Thus, the IEEE 1609.2 promotes the use of CRLs to manage revocation while assuming pervasive roadside architecture. CRLs provide privacy, as all users ask for the same file and they check the certificate status locally.

Raya *et al.* [17] propose the use of a tamper-proof device (TPD) to store the certificates. They investigated the privacy issue by proposing a pseudonym based approach using anonymous public keys and the PKI, where the public key certificate is needed, giving rise to extra communication and storage overhead. Thus, when a vehicle is compromised/misbehaving, it can be removed from the network by just revoking the TPD. To ensure that messages from this OBU are not considered valid once the certificates have been revoked, revocation information must also be distributed via CRLs. The authors also proposed to use frequently updated anonymous public keys to fulfill users' requirement on identity and location privacy. To reduce the bandwidth consumed by the transmission of CRLs, these authors proposed to compress the CRLs by using Bloom filters. However, this method gives rise to false positives which degrades the reliability of the revocation service.

Authors in [18] proposed a distributed certificate-service (DCS) scheme that introduced an aggregate batch verification technique for authenticating certificate-based signatures, which significantly decreased the verification overhead. Using DCS vehicles can update their pseudonymous certificate sets from the certificate issuer by V2I communication. Once each certificate has a short-time period and is used in a specifically geographic region, the CRL that

broadcasted in a region can decrease. However, the CRL size still depends on how many pseudonymous certificates are held by the revoked vehicles

Other proposals are based on identity-based (ID-based) signatures and group signatures to provide the revocation service. Group signature-based schemes are proposed in [19, 20, 21], where signer privacy is conditional on the group manager. As a result, all these schemes have the problem of identity escrow, as a group manager who possesses the group master key can arbitrarily reveal the identity of any group member. In addition, due to the limitation of group formation in VANETs, the group-based schemes [19, 21, 22] may not be applied appropriately. The election of group leader will sometimes encounter difficulties since a trusted entity cannot be found among peer vehicles. In [19], group signatures for OBUs and identity-based signatures for RSUs have been proposed in order to maintain security and privacy. A message received from an OBU can be verified by its signature; so that receiver can determine whether that OBU is legitimate. However, coverage of multi-hop routing is lacking in that proposal. On the other hand in [20], authors proposed an efficient conditional privacy preservation (ECPP) protocol to overcome the limitation of pre-storing a large number of anonymous certificates while preserving conditional privacy. Since a vehicle should change anonymous certificate quite often to avert tracing of messages, it should frequently interact with RSUs. This short-lived anonymous certificate needs to be sent and forwarded to verifiers for validating messages from anonymous originator.

Regarding ID-based protocols, authors in [23] proposed an ID-based security framework for VANETs to provide authentication, nonrepudiation, and pseudonymity. However, their framework is limited by the strong dependence on the infrastructure for short-lived pseudonym generation, which renders the signaling overhead overwhelming. The proposed nonrepudiation scheme enables a single authority to retrieve the identity which may raise the concern on potential abuse. Authors in [24] adopted an identity-based (ID-based) ring signature scheme to achieve signer ambiguity and hence fulfill the privacy requirement in VANET applications. The main drawback of the ring signature scheme in the VANET context, is the unconditional privacy, resulting in the traceability requirement unattainable.

Finally, some proposals in the literature divert from the IEEE 1609.2 standard and use the Online Status Checking Protocol (OCSP)[25]. OCSP is a request/response protocol between clients and responders. An OCSP responder is a trusted intermediate authority for revocation data distribution. Requests may or may not be signed by the client but all the responses must be signed so that clients can ensure that they are communicating with an authorized OCSP responder. In VANET, there is a proposal called ADOPT (Ad-hoc Distributed OCSP for Trust) [26] that provides a revocation service based on OCSP in a decentralized manner. ADOPT uses cached OCSP responses that are distributed and stored on intermediate nodes in the VANET.

3. EPA: Efficient and Privacy-Aware revocation Mechanism

3.1. Overview

EPA's main idea relies on the use of positive proofs of the certificate's no-invalidity instead of forcing vehicles to download huge revocation lists. A no-invalidity proof gives evidences that a given certificate has not been revoked. These proofs are obtained from a Merkle hash tree (MHT) that is constructed from the list of revoked certificates. A Merkle hash tree (MHT) [14] is essentially a tree structure that is built with a One Way Hash Function (OWHF). The leaf nodes hold the hash values of the data of interest (`data1`,`data2`,...) and the internal nodes hold the hash values that result from applying the OWHF to the concatenation of the hash values of its children nodes. In this way, a large number of separate data can be tied to a single hash value: the hash at the root node of the tree. MHTs can be used to provide an efficient and highly-scalable way to distribute revocation information. This MHT allows the CA to accumulate the set of revoked pseudonyms into a single value (root of the MHT) so that RSUs can efficiently compute proofs that demonstrate that a certificate has not been accumulated, i.e., a proof that a certificate is not revoked. The CAs will be in charge of managing the root value of the MHT and the CRLs that are only transmitted to the RSUs. By using the root value, any network entity will be able to check the validity of a given certificate once they obtain the corresponding no-invalidity proof. To prevent RSUs from gaining knowledge about the statistics of the PKI (i.e., who sends a message to whom, how often, etc.), each vehicle will be in charge of managing the validity proofs of the certificates they own. Vehicles contact the RSUs when they need to update their proofs and/or the root value. Hence, vehicles' privacy is preserved as they do not disclose any information when updating the revocation data.

Figure 1: EPA phases overview

Basically, EPA consists in 4 different phases (see fig. 1):

1. *Initialization*: The CA appends the identifier of any revoked certificate into a CRL. From this list, the CA calculates the root of the MHT where each leaf of the tree represents a revoked certificate. This root is signed and appended to the CRL as an extension. The extended CRL is communicated to the RSUs via a secure wireline.

2. *Repositories creation*: RSUs reconstruct the MHT

from the extended-CRL[1]. Once the MHT is constructed they can answer to any certificate status query and provide the corresponding no-invalidity proof.

3. *Certificate status testing*: Vehicles detect any entity in range and randomly choose a candidate to forward the no-invalidity proof query for a given certificate. In turn, the chosen candidate randomly chooses another candidate and forwards the query. This protocol runs until the RSU is reached and replies with the signed root value and the corresponding certificate's no-invalidity proof. Upon request from any other entity, the vehicle has to provide them with the appropriate proof. The checking entity must perform the no-invalidity test which consists in checking that the certificate does not belong to the MHT.

4. *Revocation data updating*: According to its certificate practice statement, the CA must issue a new extended-CRL periodically. These data will be transmitted to the RSUs, which in turn will transmit the new root value and the corresponding no-invalidity proofs to the vehicles.

3.2. Security Architecture

In this section we explain the security architecture necessary to support EPA and describe the details of each phase.

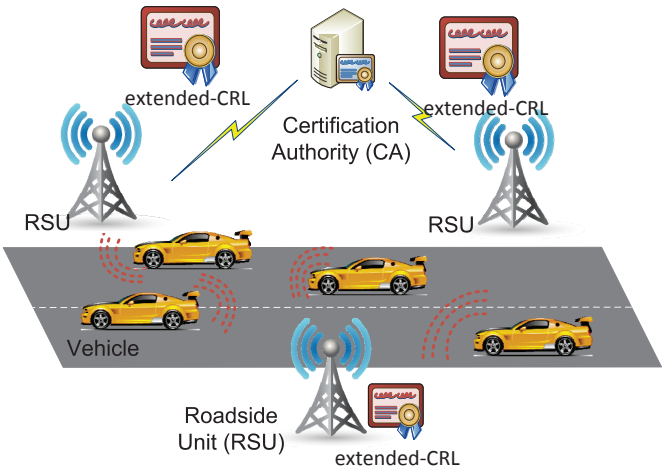

Figure 2: System Architecture.

The security architecture is an adaptation of a hierarchical PKI system to a vehicular scenario. This architecture consist of 3 different types of entities (see Fig. 2):

[1]Note that vehicles could also act as mobile repositories. However, downloading the extended-CRL will be time consuming and costly, and assuming that kind of connectivity to the infrastructure is not realistic in such disruption-prone networks.

1. *Certification Authorities*: CAs are responsible for holding and managing the credentials and identities of all the vehicles which are registered under its hood. CAs are responsible for generating the set of pseudonyms that are stored in each OBU. They are also responsible for managing the revocation information and making it accessible to the rest of the entities. By definition of TTP, the CA should be considered fully trusted by all the network entities.

2. *Road-Side Units*: RSUs are fixed entities that are fully controlled by the CA. They can access the CA anytime via wireline, which does not suffer from disconnections. If the CA considers that an RSU has been compromised, the CA can revoke it. RSUs will act as repositories of the CSI.

3. *Vehicles*: They are the clients of the network. They have their cryptographic material stored in a tamper-proof device (TPD). Vehicles access the RSU using the IEEE 802.11p.

3.3. EPA Tree

In this section, we introduce the data structure that EPA uses to handle the revocation service. In this sense, we define the EPA tree as a particular case of Merkle Hash Tree. The EPA tree is a binary hash tree where each node represents a revoked certificate.

We denote by $N_{i,j}$ the nodes within the EPA tree, where $i, j \in \{0, 1, 2...\}$ represent respectively the i-th level and the j-th node in the i-th level. We denote by $H_{i,j}$ the cryptographic (hash) value stored by node $N_{i,j}$.

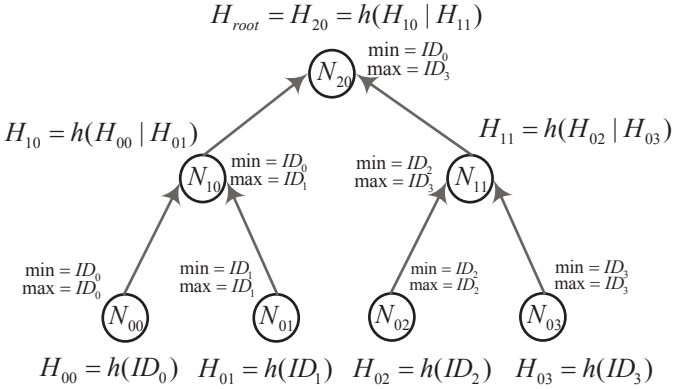

Figure 3: Sample EPA Tree.

Nodes at level 0 are called "*leaves*" and they represent the data stored in the tree. In the case of revocation, leaves represent the set Φ of certificates that are revoked at a given instant t,

$$\Phi_t = \{ID_0, ID_1, \ldots, IDj, \ldots, ID_n\}. \quad (1)$$

According to the IEEE 1609.2 Standard [1] a revoked certificate is identified in a CRL by its hash. The `CertID8` and `CertID10` types are used to identify a certificate. The

`CertIDX` for a given certificate, where X is 8 or 10, shall be calculated by calculating the SHA-256 hash of the certificate and taking the low-order X bytes of the hash output. Let ID_j be the low-order 10 bytes of the hash output of a revoked certificate. If the ID_j is stored at leaf $N_{0,j}$, then $H_{0,j}$ is computed as:

$$H_{0,j} = h(ID_j)\,, \tag{2}$$

where $h()$ corresponds to the OWHF function.

Leaves are ordered in the following way: leaves on the left represent smaller ID than leaves on the right. Each node also stores the minimum and maximum ID of its children. If a leaf has no children, then it uses its own ID for the maximum and minimum values.

To build the EPA tree, two adjacent nodes at a given level i ($N_{i,j}$, $N_{i,j+1}$) are combined into one node in the upper level, which we denote by $N_{i+1,k}$. Then, $H_{i+1,k}$ is obtained by applying h to the concatenation of the two cryptographic values:

$$H_{i+1,k} = h(H_{i,j}|\, H_{i,j+1}). \tag{3}$$

At the top level there is only one node called the "root". H_{root} is a digest for all the data stored in the EPA tree. Figure 3 shows a sample EPA tree.

Definition 1. *Let the $\mathcal{D}igest$ be the concatenation of the certification authority distinguished number DN_{CA}, the root hash H_{root} and the validity period of the CRL. Once created, the $\mathcal{D}igest$ is signed by the CA.*

$$\mathcal{D}igest = \{DN_{CA}, H_{root}, ValidityPeriod\}_{SIG_{CA}}. \tag{4}$$

Definition 2. *Let the $\mathcal{P}ath_{ID_j}$ be the set of cryptographic values necessary to compute H_{root} from the leaf ID_j.*

Note that the $\mathcal{D}$igest is trusted because it is signed by the CA and it is unique within the tree. Meanwhile, $\mathcal{P}$aths are different for each leaf. An end Entity can verify whether $ID_j \in \Phi$ if the MHT provides a response with the proper $\mathcal{P}\text{ath}_{ID_j}$ and the $\mathcal{D}$igest of the MHT. For instance, let us suppose that a certain user wants to find out whether ID_1 belongs to the sample EPA tree of Figure 3. Then,

$\mathcal{P}\text{ath}_{ID_1} = \{H_{0,0}, H_{1,1}\}$,

$\mathcal{D}igest = \{DN_{CA}, H_{2,0}, ValidityPeriod\}_{SIG_{CA}}$.

The response verification consists in checking that $H_{2,0}$ computed from the $\mathcal{P}\text{ath}_{ID_1}$, $h(h(h(ID_1)|H_{0,0})|H_{1,1})$ matches the cryptographic value $H_{2,0} = H_{root}$ included in the $\mathcal{D}$igest:

$$H_{root} = H_{2,0} = h(h(h(ID_1)|H_{0,0})|H_{1,1})\,. \tag{5}$$

Note that the EPA tree can be built by a Trusted Third Party (e.g., a CA) and freely distributed to untrusted repositories. The EPA tree cannot be forged, that is, any change in the tree made by a non-TTP will be detected. This is due to any modification in the EPA tree (for instance, the addition or deletion of a leaf node) causes H_{root} to change. As H_{root} is included in the $\mathcal{D}$igest, which is signed by the CA, this modification will cause the signature to be invalid. To perform a successful attack, the attacker would need to find a pre-image of an OWHF, which is computationally unfeasible by definition.

3.4. EPA's Operating Mode

3.4.1. Assumptions and threat model

We will make the following assumptions about the vehicular network.

- The links between vehicles and RSUs are always bidirectional.
- Vehicles and RSUS have a half-duplex constraint, i.e., they cannot simultaneously transmit and receive a message .
- An RSU is not always within the communication range of every user.
- Every wireless node has enough computation power to execute encryption and decryption algorithm.
- There is a trusted certificate authority (CA) outside the ad hoc network, which issues public key and private key to the wireless nodes inside the network.
- Each wireless node holds only one IP address for its communication in the ad hoc network, by which it will be recognized by all other wireless nodes.
- There are some nodes that are not willing to cooperate for routing and data delivering and possibly actively intent to tamper the routing protocol.
- Nodes cannot impersonate several identities at the same time. That is, we assume that there are Sybil detection and localization mechanisms that are already deployed (e.g., [27, 28, 29]).

On the other hand, we consider the threat posed by an adversary who wishes to ascertain the identity of the originator of the CSI requests. In our scenario, it is the RSU who plays the role of the privacy attacker. The users involved in the multi-hop routing protocol, however, are assumed to be partially trusted. We also assume that the attacker strives to guess the identity of the first sender (originator) of a the CSI request, knowing only the user who last forwarded it. From all this information, the RSU may estimate the most likely identity of the CSI requester, although with limited certainty. This could be interpreted as an adversary with a local view of the network, possibly due to their limited coverage range.

The immediate goal is to identify the identity of the sender of a non-invalidity proof request. Ultimate purposes include profiling of user interests and behavior inferred from statistically matching the contents of queries with sender identities, and violation of anonymity in sensitive reporting.

3.4.2. System Initialization

During the initialization, the CA creates all the necessary cryptographic material to provide the authentication and the revocation service. In this first stage, the CA creates the *extended-CRL* and delivers it to the RSUs. An *extended-CRL* is basically a standard CRL with an appended extension. This extension can be used by non-trusted entities (RSUs and vehicles inside the VANET) to act as repositories and answer to certificate status requests. All the tasks of this system initialization are performed in the CA locally, so the computational delay is not suffered by the other network entities.

The CA initializes the system by executing Algorithm 1. Note that we assume that all vehicles are equipped with a tamper-proof device (TPD).

Algorithm 1: System initialization

1 Select a generator $P \in \mathbb{G}_1$ of prime order p
2 Select a master secret key $sk_{CA} \in \mathbb{Z}_p^*$
3 Set the corresponding public key $pk_{CA} = sk_{CA} \cdot P$
4 Given the set of n elements $\mathcal{R} = \{ID_1, ID_2, \ldots, ID_n\}$ of revoked certificates, create the CRL including all the elements
5 Compute the MHT as explained in Sec. 3.3
6 Compute and sign the $\mathcal{D}igest$ from the MHT's root.
7 Append the $\mathcal{D}igest$ to the CRL and sign it
8 **forall** OBU_k, CA **do**
9 Generate a set of anonymous certificates (Υ) $pseud_k = \{cert_i^k(ID_i^k, PK_i^k), sig_{CA}(ID_i^k||PK_i^k))|1 \leq i \leq \alpha\}$
10 Upload $pseud_k$ in TPD_k of OBU_k
11 Convey the extended-CRL to the RSUs, and publicize the $\mathcal{D}igest$ to all the OBUs

It should be noted that in step (1), PK_i denotes the i^{th} public key for OBU_k , where the corresponding secret key is SK_i; ID_i denotes the ith pseudonym for OBU_k, where the CA is the only entity that can relate ID_i to the real identity of OBU_k; and α is the number of pseudonyms loaded in each OBU. After this first stage of System Initialization, the RSUs have a copy of the *extended-CRL*, which contains exactly the same CSI than a standard CRL and it is valid for the same time. The advantage of an *extended-CRL* is that any non-trusted entity in possession of it can generate again the EPA tree locally, and obtain the root hash. As the *extended-CRL* also includes the $\mathcal{D}$igest, which is signed by the CA, this entity has an authenticated version of the EPA tree and can answer to CSI requests in an off-line way.

3.4.3. Repositories Creation

After the initialization phase, any vehicle possesses a set of pseudonyms to preserve their privacy. They can obtain the $\mathcal{D}$igest from any RSU in range. To become a repository an entity must follow the next steps:

1. The entity obtains the *extended-CRL* either from the CA or from another entity that has an up-to-date copy of the *extended-CRL* in its cache. Notice that the CA uses a secure wireline to communicate with the RSUs, while the RSUs use a wireless link to communicate with the vehicles.

2. Once the *extended-CRL* has been downloaded, the entity verifies that the signature of the *extended-CRL* is valid and corresponds to the CA. If so, the entity generates locally the EPA tree using the serial numbers within the *extended-CRL* and following the same algorithm than the CA (as explained in Section 3.3). The root hash of the tree created from the *extended-CRL* entries must match the signed root value contained in the $\mathcal{D}$igest.

3. At this moment, the entity can create any certificate no-invalidity proof until the *extended-CRL* expires.

3.4.4. Certificate Status Testing

After the second stage, RSUs will be able to act as repositories. The last stage of the mechanism consists in providing the certificate status information to any vehicle that needs to obtain a no-invalidity proof of their certificates. The protocol for status information exchange is based on the hash tree structure and it allows checking the integrity of a single *extended-CRL* entry with only some hash material plus the $\mathcal{D}$igest (included in the extension). On the one hand, this is much more efficient than broadcasting the entire *extended-CRL*. On the other hand, the mechanism is fully offline (the only trusted authority is the CA), which is a very good feature because sometimes it may be impossible for vehicles to reach the CA due to lack of coverage.

However, if a vehicle forwards directly the no-invalidity proof requests to the RSU, it could be easily tracked as the RSU will know which pseudonyms corresponds to the requesting vehicle. To avoid this traceability, EPA uses an anonymity protocol based on Crowds [16].

A. EPA's probabilitic routing

EPA builds on top of a generic multi-hop routing protocol. The selection of this protocol will determine the performance of our approach, both in terms of QoS and user anonymity. EPA is deployed on top of this routing protocol, and operating at the application layer.

To exemplify EPA's performance, we deployed our mechanism over the adaptive Secure Ad hoc On-demand Distance Vector (A-SAODV) routing protocol [30, 31]. e use A-SAODV in its unicast version where routing decisions are made using distance vectors. Note that the number of participants basically depends on AODV, the forwarding probability p and the maximum delay of the query. Each non-validity proof request is sent in a different message and thus a different path. Each node handling a CSI request gives a probability p to the shortest path determined by

the A-SAODV protocol to the closest RSU, whereas the probability to choose another path is then $1-p$. Thus, each of the $N-1$ neighbor vehicles will have a probability of $\frac{1-p}{N-1}$ of being selected as the next forwarding node.

Figure 4: Fordwaring routing protocol

Algorithm 2: Probabilistic routing protocol

Input: SourceID, *RSU*, timestamp
Output: NextHopNodeID

```
if CurrentNodeID != RSU then
    SourceID ← CurrentNodeID
    Find shortest path P* to RSU using A-SAODV
    if timestamp < maxDelay then
        if rand() ≤ p then
            NextHopNodeID ← Next node in P*
        else
            NextHopNodeID ← Any other neighbor
    else
        NextHopNodeID ← Next node in P*
```

Figure 4 shows an example of this anonymous protocol. In this example, a vehicle to validate three of its pseudonyms (i.e., $pseud_a$, $pseud_b$ and $pseud_c$) forwards three no-invalidity proof requests to the RSU following three different paths. In this case, the shortest direct path to the RSU using AODV is only used for $pseud_c$. Note that EPA does not necessarily use the same path to get the non-invalidity proofs for the same vehicle.

As described in Alg. [**?**], to limit the packet delay that can be induced by potential loops, we use a timeout that will bound the maximum delay. Once this timeout is reached, the vehicle that is currently managing the CSI request will send the message directly to the RSU in range. This timeout is based on the timeout algorithm defined in [32] which takes into account the transmission range and the vehicles velocity.

To forward the non-invalidity proof to the requesting vehicle, it is necessary to establish a backward path. EPA follows the same strategy that the original Crowds mechanism,i.e., to add a `path_id` field to determine a node in a path. The `path_id` field constitutes an overhead of 128 bits [16].

Taking advantage of the `RERR` message defined in AODV, it is possible to notify that a breakage in the path has occurred and some reply messages will be lost. Of course as the identity of the sender remains anonymous, is only possible repair the path toward destination. In other words the reply mechanism is intermediate node presence dependent.

B. Obtaining non-invalidity proofs

To obtain the non-validity proofs of its pseudonyms a vehicle must follow the protocol described in Algorithm 3. Contrary to traditional certificate status checking mechanisms, vehicles do not need neither to download the whole CRL nor to disclose the ID of the certificate they need to check. Note that in each update the OBU has to choose k pseudonyms from the set of valid pseudonyms it has previously stored in its TPD. Once k pseudonyms are chosen, the vehicle must obtain the corresponding no-invalidity proof following the aforementioned protocol. Thus, the vehicle just obtains the right amount of no-invalidity proofs it will need during the lifetime of the $\mathcal{D}igest$. Haas, Hu, and Laberteaux in [6] recommend changing pseudonyms every 10 minutes while driving 2 hours per day. Traditionally, new updates of the CSI are published each 24 hours, so this means downloading 12 validity proofs every day. Note that the size of tens of validity proofs is still much lower than the size of the CRL that contains thousands of revoked certificates. Therefore, EPA not only improves the privacy of the users but also the bandwidth costs required to check the validity of any certificate.

Algorithm 3: Obtaining the Certificates' No-invalidity Proofs

Input: Set of pseudonyms (Υ)
Output: No-invalidity Proofs

```
if Digest not cached or Expired Validity Period in Digest or not valid signature then
    Download Digest from any repository in range.
else
    Randomly select k pseudonyms from the set Υ
    forall Selected pseud_i do
        Hash the pseud_i and take the low-order 10 bytes of each hash output.
        Obtain the corresponding Paths from the RSU (using the aforementioned anonymous protocol)
        Verify that the H_root calculated from the Paths matches the H_root contained in the Digest.
return Set of Paths (No-invalidity proofs)
```

First of all, the OBU must check that each $\mathcal{P}ath$ included in the response is correct, that is, that the `rootHash` computed from the $\mathcal{P}$ath matches the `rootHash` included in the $\mathcal{D}$igest. However, it is worth noting that testing the validity of a path is not enough if a target certificate has not been revoked. Additionally, the OBU also needs to ensure that the $\mathcal{P}$aths provided belong to real adjacent nodes (remember that the repository is a non-TTP, so the user can be misled into believing that a certain pair of nodes within the tree are adjacent leaves).

Notice that as the H_{root} is signed by the CA, it is just as impractical to create falsified values of the $\mathcal{P}$ath as it

Algorithm 4: Algorithm to calculate the $\mathcal{P}$ath of a given certificate.

Input: SN_{target}
Output: $\mathcal{P}$ath

```
N_ij = root;
while N_ij.max ≠ N_ij.min do
    i = i − 1
    j = 2 · j
    if N_ij.max < SN_target then
        Path.add(N_ij)
        j = j + 1
    else
        Path.add(N_i,j+1)
return Path
```

is to break a strong hash function. In case the certificate is not revoked, the repository sends the adjacent leaves to the requested certificate. In this respect, the repository has to prove that a certain certificate (ID_{target}) does not belong to the set of revoked certificates (Φ). Recall that ID_j denotes the low-order 10 bytes of the hash output of a revoked certificate. To prove that $ID_{target} \notin \Phi$, as the leaves are ordered, it is enough to demonstrate the existence of two leaves, a minor adjacent (ID_{minor}) and a major adjacent (ID_{major}) that fulfill:

1. $ID_{major} \in \Phi$.
2. $ID_{minor} \in \Phi$.
3. $ID_{minor} < ID_{target} < ID_{major}$.
4. ID_{minor} and ID_{major} are adjacent nodes.

Next, we describe a recursive algorithm that given a certain couple of $\mathcal{P}aths$, verifies whether they actually belong to "real" adjacent leaves. The algorithm works without adding any extra information to the protocol or the data structures. The alleged adjacent leaves are denoted by $N_{0,j}$ and $N_{0,j+1}$.

It must be pointed that the strength of the above algorithm resides in the position that a certain node occupies relative to its father, in other words whether a certain node is LEFT or RIGHT. Notice that the end user can trust this information since the relative node positions cannot be swapped by a malicious repository because we use a non-commutative hash function. If the malicious repository modifies the concatenation order, then it changes the cryptographic value of the next step (6)

$$H_{i+1,k} = h(H_{i,j} | H_{i,j+1}) \neq h(H_{i,j+1} | H_{i,j}) . \quad (6)$$

After executing Algorithm 3, the vehicle has the no-invalidity proofs necessary to demonstrate to any other entity that the pseudonyms it is using are valid. When a vehicle needs verifying the validity of a given pseudonym it must:

Algorithm 5: Algorithm to test the adjacency of two nodes from their $\mathcal{P}$aths.

Input: Conresponding $\mathcal{P}aths$ for $N_{0,j}$ and $N_{0,j+1}$
Output: `Adjacent`

```
i = 0
while Common father between N_0,j and N_0,j+1 is not found do
    Calculate H_i+1,m H_i+1,n, i.e., cryptographic height 0.
    values of the fathers of N_i,j and N_i,j+1.
    if N_i,j = N_i+1,m.left and
    N_i,j+1 = N_i+1,m.right then
        Adjacent = TRUE
        break
    else
        Adjacent = FALSE
        break
    if N_i,j ≠ N_i+1,m.right then
        Adjacent = FALSE
        break
    else if N_i,j+1 ≠ N_i+1,n.left then
        Adjacent = FALSE
        break
    else
        i=i+1
return Adjacent
```

1. Obtain the no-invalidity proof from the corresponding vehicle, i.e., the $\mathcal{P}$aths of the corresponding ID_{minor} and ID_{major}.
2. Obtain the hash of the pseudonym and take the low-order 10 bytes.
3. Verify that the value of these 10 bytes is between ID_{minor} and ID_{major} and that ID_{minor} and ID_{major} are adjacent nodes.
4. Verify that the H_{root} calculated from the corresponding $\mathcal{P}$aths matches the H_{root} contained in the $\mathcal{D}$igest.

Note that sending a new request for a proof of non-invalidity for the same pseudonym does not necessarily lead to a loss of anonymity. The requesting vehicle could be traced only in the case that it sends the same request to the same RSU and its neighbors are completely different from the first time it sent the same request. Thus, CSI updating policies must be programmed at the OBU so that this situation is avoided. These policies include:

1. Requesting non-validity proofs only when several vehicles are under the coverage of the target repository.
2. Do not request non-validity proofs to a repository that already has provided these proofs for the same pseudonyms previously.

If these rules are enforced, OBUs could update non-validity proofs even when they have already used those pseudonyms.

3.5. Security Analysis

In this subsection, we analyze the security of the proposed scheme in terms of message authentication and integrity, nonrepudiation, and privacy preserving.

1. *Privacy preserving.* EPA avoids that private information could be disclosed during the revocation process. When obtaining/updating the pseudonyms' no-invalidity proofs, vehicles use an anonymous protocol so that the repositories cannot link the query with the identity of the originator. Thus, vehicles just ask for the status of their own certificates.

2. *Mis-authentication Resistance*: In the proposed scheme, each entity should provide the no-invalidity of the pseudonym it is using. This proof is directly linked with the signed-root value of the MHT. Therefore, if the hash of pseudonym lies in one of the leaves of the MHT, the message will be dropped.

3. *Non-repudiation.* Based on the signature enclosed in the $\mathcal{D}$igest, vehicles can reveal the identity of the CSI issuer, while the issuer cannot deny that the message was generated by itself.

4. *Replay attack resilience*: Consider an adversary $\mathcal{A}$ whose certificate r_i has been revoked. Since the $\mathcal{D}igest$ issued by the CA includes the current time stamp T_i, $\mathcal{A}$ cannot use a validity proof valid at time T_i and replay it at a later time T_{i+1} to pass the revocation checking process as the receiving OBU compares the current time T_{i+1} with that included in the current $\mathcal{D}igest$. Consequently, EPA is secure against replay attacks.

4. Performance Evaluation

In this section, we evaluate the efficiency of EPA and we compare it with other certificates status management protocols designed for VANET.

4.1. Analytical Evaluation

4.1.1. Storage Overhead of Vehicles

In our scheme, the pseudonym set costs storage space in vehicles. Through a requesting process, a vehicle gets n pseudonyms and k validity proofs with $k < n$. Let S_{cert} denote the size of a certificate, and S_{proof} denote the size of a no-invalidity proof. The size of the certificates is fixed while the size of the proof basically depends on the height of the MHT that in turn depends on the number of revoked certificates. Then, the storage for pseudonyms and their validity proofs is:

$$Stor = n \times S_{cert} + k \times S_{proof}$$

According to 1609.2 standard [1], we employ ECDSA as the basic signature algorithm [33], so that $S_{cert} = 200$ Bytes. According to NIST statistics [34] 10% of the certificates need to be revoked during a year and assuming that a regional certification authority could manage around 50,000 vehicles, the size of the proof will be of around 700 Bytes. Then, $Stor \approx 109$ MBytes. This storage is affordable for today's devices.

4.1.2. Revocation Overhead

Updated revocation data must be available to any vehicle. In EPA, vehicles just need to download the $\mathcal{D}igest$ and the corresponding no-invalidity proofs, while current proposals such as DCS [18] and ECPP [20] need to download the whole CRL. Table 1 presents the CRL size to revoke one vehicle.

Mechanism	Unit size	Item number	Total size (bytes)
Std 1609.2[1]	21 bytes	$\frac{L_w+1}{2}$	10.5*(L_w+1)
DCS [18]	8 bytes	$\frac{L_w+1}{2}$	4*(L_w+1)
ECPP [20]	21 bytes	$\frac{L_w+1}{2}$	10.5*(L_w+1)
ADOPT [7]	586 bytes	1	586
EPA	710 bytes	1	710

Table 1: Comparison of the overhead introduced by EPA and other certificate validation mechanisms.

In ECPP, and DCS, all the pseudonyms of unexpired certificates belonging to the revoked vehicle should be added into the CRL. If the maximal size of the short-time pseudonymous certificate set in both ECPP and DCS is L_w, the average number of unexpired certificates is $(L_w + 1)/2$. On the other hand in EPA, the new revoked pseudonyms just have to be added to the $\mathcal{D}igest$. To update the revocation data, vehicles just have to download this $\mathcal{D}igest$ and k validity proofs. Thus, the size of the data to update the revocation information in EPA is lower than in DCS or ECPP as it does not grow linearly with the number of revoked certificates. Note that EPA introduces the second lowest total overhead in the network when downloading the revocation data. It is only improved by ADOPT that is an online revocation mechanism that assumes the continuous availability of the revocation infrastructure.

4.1.3. Privacy and traceability

The basic idea behind EPA is very simple: instead of querying by a specific certificate serial number that belongs to another vehicle, the current vehicle queries several serial numbers k where k is the cardinal of a subset of its pseudonyms. Thus, EPA achieves to effectively hide the identity of the vehicle they want to communicate with. The only data divulged to the RSU (third party) are the k pseudonyms that belong to a single vehicle. However, as

EPA takes advantage of an anonymous forwarding protocol to obtain the no-invalidity proof, a compromised RSU cannot track the identity of a vehicle. Hence the tracking capabilities of a compromised RSU remain the same that any other entity.

On the other hand, the schemes that adopt RSU-aided revocation mechanisms such as DCS and ECPP cannot achieve conditional anonymity against the RSUs. For instance, in ECPP, a vehicle validates the status of its certificates from an RSU by its invariable credential. Therefore, when the service records stored in an RSU are leaked, the adversary can find out all the certificates that the RSU has issued for the interested vehicle. In DCS, a vehicle obtains the RSU service by a pseudonymous certificate issued by the other RSUs. This way, the adversary does not know which vehicle requests the service, but it can correlate the pseudonymous certificates belonging to the same vehicle. Here, we develop a probabilistic model to analyze the risk that the knowledge of an RSU is used to track an interested vehicle.

Suppose there is a compromised RSU, and α vehicles in a certain region. This compromised RSU gathers some traffic routine messages during μ time slots and tries to analyze the mobile route of an interested vehicle. In each time slot, the RSU has recorded the certificates used by these vehicles. Let $Pr(\phi)$ denote the probability that the RSU distinguishes the pseudonymous of the interested vehicle from ϕ candidate certificates, where $Pr(\phi) = \frac{1}{\phi}$. If the RSU can correlate μ certificates of the interested vehicle, the tracing analysis succeeds. Let P_{succ} denote the success traceability probability. When the RSU is in secure state, the adversary has to find out every certificate of the interested vehicle from α certificates at each time slot. Therefore, $P_{succ} = \frac{1}{\alpha^{\mu}}$. In EPA, when the RSU is compromised, the adversary can get no useful information; thus, $P_{succ}^{EPA} = \frac{1}{\alpha^{\mu}}$. In ECPP, the adversary can directly find out the pseudonymous certificates of the interested vehicle; thus, $P_{succ}^{ECPP} = 1$. In DCS, the adversary has to confirm just one certificate of the interested vehicle; thus, $P_{succ}^{DCS} = \frac{1}{\alpha}$. I we set the number of time slots that the compromised RSU can monitor equal to ten (i.e., $\mu = 10$), Fig. 5 plots the success traceability probability versus number of vehicles in the region. It can be seen that EPA provides the best privacy preservation.

Figure 5: Success traceability probability when the RSU is compromised

Finally, note that EPA not only prevents the tracking of any vehicle during the revocation service, but also prevents that a compromised RSU could know who is communicating with who. Let us define *achievable privacy* as the probability of guessing the target certificate. Online certificate status checking mechanisms such as OCSP or ADOPT have an achievable privacy equal to 1, i.e., a compromised RSU can exactly know who are the entities involved in any communication. On the other hand, other approaches such as CRL, DCS and ECPP obtain the maximum achievable privacy. Similarly, EPA also achieves the same level of privacy than the CRL-based mechanisms. Thus, while EPA requires less revocation overhead, it reaches the same level of privacy than traditional revocation mechanisms.

4.1.4. Certificate Status Checking Delay

We compare the message status validation delay employing the IEEE 1609.2, ECPP, DCS and ADOPT with that employing EPA to check the revocation status of an OBU. This delay corresponds to amount of time necessary to check the validity of a given certificate once the revocation data has been retrieved. The IEEE 1609.2 trial use standard proposes the use of CRL to check the status of a certificate. To check the validity of certificate against the CRL, a progressive search of the revoked certificates is performed. Let T_{hash} and T_{mul} denote the time required to perform a hash operation and a point multiplication, respectively. The elliptic curve digital signature algorithm is the digital signature method chosen by the VANET standard IEEE1609.2, where a signature generation takes T_{mul} and a signature verification takes $4\,T_{mul}$. In EPA, to verify a credential, a verifier must perform hash operations to compute the current contents of the leaf node corresponding to the target vehicle's identifier. Therefore, the total computation overhead when checking the status of a certificate is $T_{hash}(logN + 1) + 4\,T_{mul}$. In [35], T_{mul} is found for an MNT curve with embedding degree $k = 6$ that is equal to 0.6 ms. In our simulation, we use the results of for real tamper-proof devices [36]. Note that these results were obtained using a PC machine Pentium III 800MHz with 256MB RAM. According to these results, the maximum performance that Pentium III can achieve for SHA-1 hashes is 2.52 μops/cycles.

Figure 6 shows a comparison between the verification delay per message using different certificate status checking mechanisms vs. the number of the revoked certificates. It can be seen that the delay using the CRL checking process increases with the number of revoked certificates. ECPP and DCS present similar delays to the ones obtained with the traditional CRL. With EPA the delay also increases but in a logarithmic manner. On the other hand, this delay remains almost constant when using ADOPT and EPA. ADOPT performs slightly better than EPA as it only requires to check the correctness of a signature and the freshness of a timestamp, while EPA also requires to check the $\mathcal{D}igest$. However, the main delay in ADOPT is not due to the verification process but to the location of a valid pre-signed response.

4.2. Simulation

In this section, we use the OMNeT++ network simulator [37] and its *INET Framework* [38] extension to evaluate the performance of EPA. OMNeT++ is a well-known discrete event simulator based on C++ which offers excellent capabilities for protocol and network modeling. The

Figure 6: Verification Delay

Figure 7: City of Barcelona. Data converted from OpenStreetMap format to SUMO format.

INET Framework [38] is a collection of protocols for the use with OMNeT++ which, among others, contains implementations of IPv4, IPv6, TCP, UDP and several application models as well as link-layer models of PPP, Ethernet and 802.11. Especially its sophisticated implementation of 802.11 Medium Access Control (MAC) and MAC Layer Management makes the INET Framework a good choice for the simulation of 802.11p-based car-to-x communication.

We use the traffic simulator SUMO (Simulation of Urban MObility) [39] in order to generate our mobility models. In order to have a basis for the analysis's of EPA, a realistic scenario was developed. The scenario comprises an urban area of 90km^2 of the Spanish city of Barcelona. RSUs are placed every 1,500 meters covering the whole city, similar to an access point in traditional wireless ad hoc networks to provide necessary infrastructure support for network setup and communications. In Fig. 7 the scenario is depicted, which shows that residential zones, small industrial areas and also parks with vegetation are existent in the scenario. A typical residential zone with many small side roads is located in the south-eastern part of the city. Throughout the simulation area, there are major roads and in the north-west as well as in the north-east of the area there are two major traffic junctions. Different streets like one-way and two-way streets with a different number of lanes as well as different speed limits of 30 or 80 km/h are included.

The main parameter settings used in the simulations are listed in Table 2. Note that we have configured our simulation to use the Nakagami propagation model. We choose this propagation model because empirical research studies have shown that a fading radio propagation model, such as the Nakagami model, is best for simulation of a vehicular environment [40].

Parameter	**Value**
Area	90 km^2
Number of RSUs	20
RSU Transmission power	28.8dBm
MAC	IEEE 802.11p
Propagation model	Nakagami
Transport protocol	UDP

Table 2: Parameter values for the reference scenario.

The transmitter and receiver powers of the OBUs were defined in the IEEE WAVE family of standards, so as receivers they have a sensitivity of -82.0 dBm (see Table 3). The MAC retransmission policies for unicast messages are the default. We use the 802.11 short retry limit of 7 retransmissions before it gives up on transmitting a given message. If the message is not transmitted after 7 retries it is dropped and it is the responsibility of the executing application to ensure that it resends the message if that is still needed. We generate network traffic using greedy UDP. By "greedy" we mean that the source node of a flow will transmit data as many as it can. In our simulation scenario, each RSU sets up a greedy UDP flow to each vehicle. Recall that UDP is a transportation layer protocol that simply transmits data down to the MAC-layer; thus, the throughput obtained by all greedy UDP flows of all

the vehicles are equivalent to the MAC-layer throughput that can be obtained by the nodes minus the bandwidth overheads introduced by MAC-layer, network-layer, and transport-layer headers. For convenience, we use aggregate UDP throughput to evaluate the 802.11(p) protocol. Since the revocation data are deemed to be very important for the robustness of the network, they are sent with the maximum priority, which corresponds to access category 3. The configuration parameters of the vehicles are shown in Table 3) .

Parameter	Value
Speed	{10,12,14,16,18,20} m/s
Max. Acceleration	5 m/s
Max. Deceleration	3 m/s
Channel bandwidth	10 MHz
OBU receiver sensitivity	-82.0dBm
OBU antenna height	1m
Type of antenna	Omnidirectional

Table 3: Car Profile.

4.2.1. Anonymity

Let us define a new metric, named *anonymity* as the probability that the a given attacker could not guess the originator of the CSI query, i.e., the attackers guess of the source node ($\hat{S}$) does not match the actual source node (S) :

$$anonymity = Prob\{\hat{S} \neq S\} = \frac{N - 1 - p(N - c - 2)}{N - 1 + p} \quad (7)$$

where N is the number of neighbors of the sender node, c is the number of malicious nodes and p is the forwarding probability.

Figure 8: Anonymity vs. mean number of hops

As shown in fig. 8 the anonymity increases with the number of hops involved in the probabilistic routing. In other words, with higher density of vehicles more vehicles are involved in the forwarding of the non-invalidity proofs. Thus, when more vehicles are present in the network it becomes more probable to have higher number of hops and hence more anonymity. Obviously, if we use a single hop routing to get the non-invalidity proof, the requesting vehicle is not longer anonymous. It is also worth noting that some packet are lost and thus, EPA's does not reach the theoretical maximum anonymity.

4.2.2. End-to-end Delay

To further evaluate EPA, we analyze the end-to-end delay, which is defined as the time to transmit a message from the sender to the receiver. That is, it includes not only the transmission time of the message, but also the time to authenticate the message. In EPA, authenticating a message consists in verifying the validity of the signature and checking that the corresponding certificates have not been revoked.

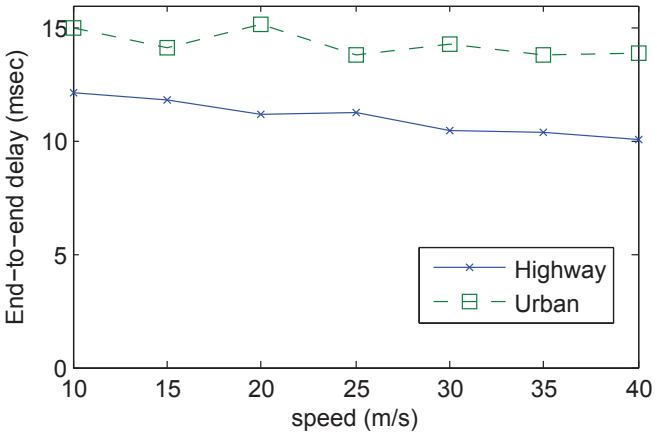

Figure 9: EPA end-to-end delay vs vehicle's speed, for different probabilities of the anonymous protocol.

Figure 9 shows the end-to-end delay in milliseconds vs. the vehicle's speed, by employing authentication using EPA. In the simulation, we consider 30,000 revoked certificates and a message payload of 10 bytes. It can be seen that the end-to-end delay decreases with the speed because the number of the received packets decreases (as well as the OBUs density) resulting in shorter waiting time for the packets to be processed by the application layer in each OBU. It can be seen that the end-to-end delay the end-to-end delay tends to be constant no matter the vehicle speed. This is mainly due to the existence of high OBUs densities and the number of received packets reaches the maximum number of packets an OBU can verify within a specific duration. However, as expected, the delay increases with the increase of the probability of forwarding the no-invalidity proof request to other entities rather than to the closest RSU. Note that with $p = 1$ means that all the proof request are directly forward to the RSU, which compromises the anonymity of the vehicle but achieves the minimum delay. In any case, the end-to-end delay is below the 110 ms, which allows to rapidly checking the status of a given certificate, while downloading a CRL containing the IDs of 30,000 revoked certificates could take minutes.

4.2.3. Message Loss Ratio

According to DSRC, each vehicle should disseminate a traffic message every 300 ms. The average message loss ratio is defined as the average ratio between the number of messages dropped every 300 ms, due to the message certificate status checking delay, and the total number of messages received every 300 ms by an OBU. Figure 10 shows the simulated message loss ratios every 300 ms for

several certificate status checking mechanisms. It can be seen that ADOPT performs the best due to the lowest authentication overhead, and the performance of EPA is almost close to ADOPT. At the initial stage of simulation, the vehicles in ECPP and DCS have no idea on which certificates are veritable and have to verify both of the certificate and the message signature for the received messages. They cannot afford so much overhead, and some messages will be dropped. In ECPP, as the number of verified certificates increases in the following stages of simulation, the message verification overhead is only contingent upon the signature verification. Because the certificate verification cost is high in DCS, the message lost ratios in them do not monotonously decrease. Moreover, we can observe that DCS also does not work well because batch verification is not efficient once the fake messages exist.

Similarly, under IEEE 16909.2, vehicles using CRLs have no idea on which certificates are veritable and have to verify both of the certificate and the message signature for the received messages. They have to download the whole CRL but they cannot afford so much overhead, and some messages are dropped. Thus, the message loss ratio is large at the beginning and decreases during the running of the simulation. EPA does not present this transient state during the initialization phase as the $\mathcal{D}igest$ can be downloaded in milliseconds.

5. Conclusions

We have proposed EPA for VANETs, which enhances the certificate status checking process by replacing the time-consuming CRL with a fast revocation checking process employing a Merkle Hash Tree. EPA not only satisfies the security and privacy requirements of VANETs but can also significantly reduce the revocation cost. Moreover, EPA enhances the driver's privacy so that the adversaries cannot trace the communication of legitimate vehicles, although they have compromised the RSUs. Taking advantage of an anonymous protocol, vehicles do not have to contact directly the RSU during the revocation service. In contrast, our anonymous no-invalidity proof querying protocol depends on a forwarding probability that determines whether the next query is forwarded to a random neighbor, for better anonymity, or the query is sent directly to the RSU, for minimum delay.

Analytic results show that allocating a small bandwidth is enough to ensure that vehicles can download fresh revocation information within few milliseconds. The performance improvement is obtained at expenses of using positive certificate status information, where vehicles have to obtain a proof of the certificate's no-invalidity prior to validate its status. Therefore, EPA significantly reduces the complexity of certificate management and achieves great efficiency and scalability, even when it is deployed in vehicular networks.

Acknowledgment

This work is funded by the Spanish Ministry of Science and Education under the projects CONSOLIDER-ARES (CSD2007-00004) and TEC2011-26452 "SERVET", FPU grant AP2010-0244, and by the Government of Catalonia under grant 2009 SGR 1362.

References

[1] IEEE, IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages, IEEE Std 1609.2-2006 (2006) 1–117.

[2] M. Nowatkowski, H. Owen, Certificate revocation list distribution in VANETs using most pieces broadcast, in: IEEE SoutheastCon 2010 (SoutheastCon), 2010, pp. 238–241.

[3] J. Haas, Y.-C. Hu, K. Laberteaux, Efficient certificate revocation list organization and distribution, Selected Areas in Communications, IEEE Journal on 29 (2011) 595 –604.

[4] A. Wasef, X. Shen, MAAC: Message authentication acceleration protocol for vehicular ad hoc networks, in: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, 2009, pp. 1 –6.

[5] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, J.-P. Hubaux, Certificate Revocation in Vehicular Networks, Technical Report, EPFL, 2006.

[6] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, J.-P. Hubaux, Secure vehicular communication systems: design and architecture, Communications Magazine, IEEE 46 (2008) 100 –109.

[7] K. Papapanagiotou, G. F. Marias, P. Georgiadis, A Certificate Validation Protocol for VANETs, 2007 IEEE Globecom Workshops (2007) 1–9. doi:10.1109/GLOCOMW.2007.4437825.

[8] A. Wasef, X. Shen, EDR: Efficient Decentralized Revocation Protocol for Vehicular Ad Hoc Networks, Vehicular Technology, IEEE Transactions on 58 (2009) 5214 –5224.

[9] C. Gañán, J. L. Muñoz, O. Esparza, J. Mata, J. Hernández-Serrano, J. Alins, COACH: COllaborative certificate stAtus CHecking mechanism for VANETs, Journal of Network and Computer Applications (2012).

[10] A. Wasef, X. Shen, EMAP: Expedite Message Authentication Protocol for vehicular ad hoc networks, Mobile Computing, IEEE Transactions on PP (2011) 1.

[11] J. Solis, G. Tsudik, Simple and flexible revocation checking with privacy, in: Proceedings of the 6th international conference on Privacy Enhancing Technologies, PET'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 351–367.

[12] Y. Sun, R. Lu, X. Lin, X. Shen, J. Su, An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications, Vehicular Technology, IEEE Transactions on 59 (2010) 3589 –3603.

[13] Y. Sun, R. Lu, X. Lin, X. Shen, J. Su, An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications, Vehicular Technology, IEEE Transactions on 59 (2010) 3589 –3603.

[14] R. Merkle, A certified digital signature, in: Advances in Cryptology (CRYPTO89). Lecture Notes in Computer Science, 435, Springer-Verlag, 1989, pp. 234–246.

[15] J. Forné, J. L. Muñoz, O. Esparza, F. Hinarejos, Certificate status validation in mobile ad hoc networks, Wireless Commun. 16 (2009) 55–62.

[16] M. K. Reiter, A. D. Rubin, Crowds: anonymity for web transactions, ACM Trans. Inf. Syst. Secur. 1 (1998) 66–92. doi:10.1145/290163.290168.

[17] M. Raya, J.-P. Hubaux, The security of vehicular ad hoc networks, in: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks, SASN '05, 2005, pp. 11–21.

[18] A. Wasef, Y. Jiang, X. Shen, DCS: An Efficient Distributed-Certificate-Service Scheme for Vehicular Networks, Vehicular Technology, IEEE Transactions on 59 (2010) 533 –549.

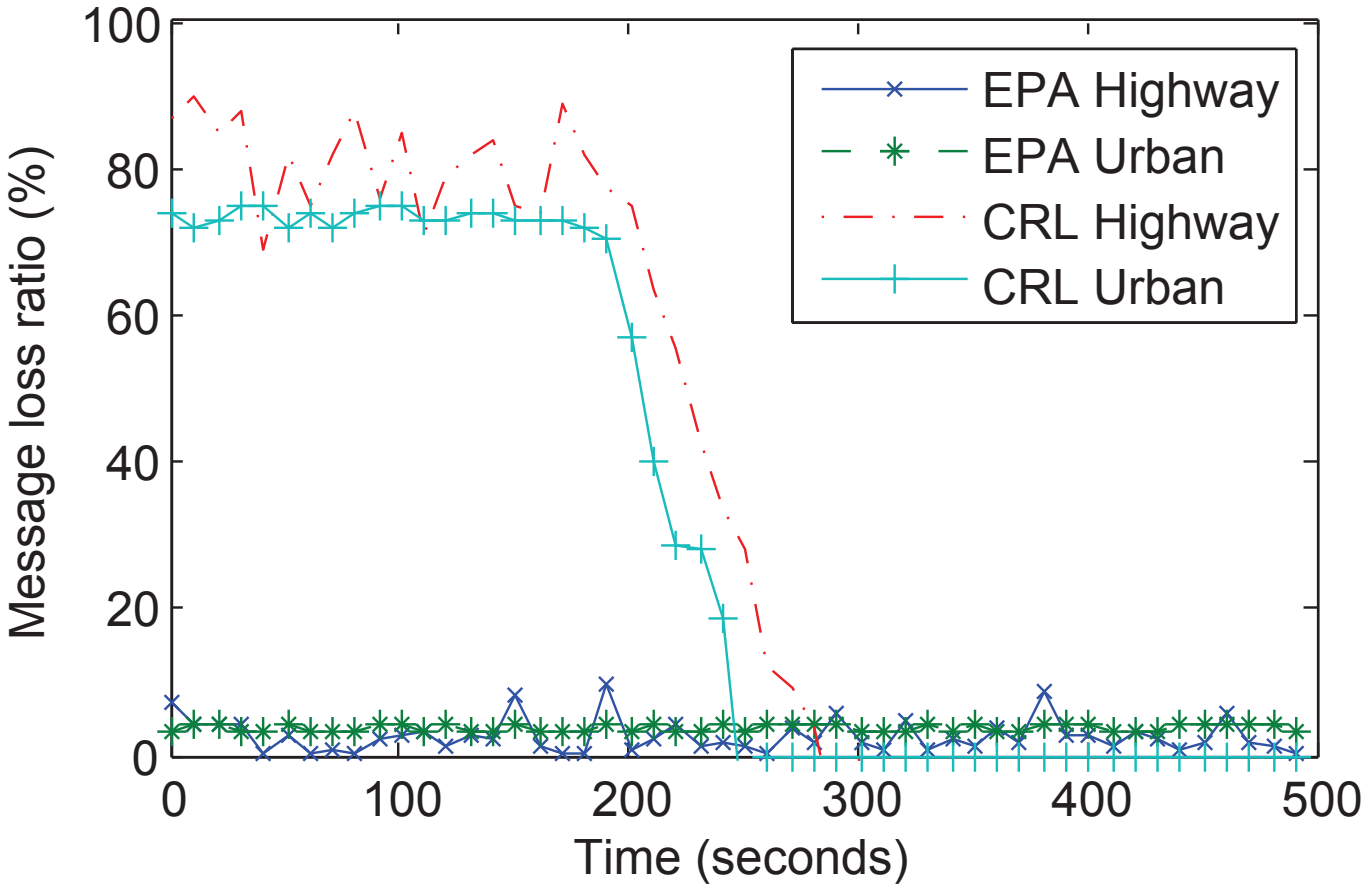

Figure 10: Comparison between message loss ratios for different schemes.

[19] X. Lin, X. Sun, P.-H. Ho, X. Shen, GSIS: A secure and privacy-preserving protocol for vehicular communications, Vehicular Technology, IEEE Transactions on 56 (2007) 3442 –3456.

[20] R. Lu, X. Lin, H. Zhu, P.-H. Ho, X. Shen, ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications, in: INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, 2008, pp. 1229 –1237.

[21] A. Studer, E. Shi, F. Bai, A. Perrig, TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs, in: Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on, 2009, pp. 1 –9.

[22] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, K. Sezaki, CARAVAN: Providing location privacy for VANET, in: in Embedded Security in Cars (ESCAR), 2005.

[23] P. Kamat, A. Baliga, W. Trappe, Secure, pseudonymous, and auditable communication in vehicular ad hoc networks, Security and Communication Networks 1 (2008) 233–244.

[24] C. Gamage, B. Gras, B. Crispo, A. S. Tanenbaum, An Identity-based Ring Signature Scheme with Enhanced Privacy, in: Securecomm and Workshops, 2006, 2006, pp. 1 –5.

[25] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, RFC 2560 (Proposed Standard), 1999. URL: `http://www.ietf.org/rfc/rfc2560.txt`, updated by RFC 6277.

[26] G. F. Marias, K. Papapanagiotou, P. Georgiadis, Adopt. a distributed ocsp for trust establishment in manets, 11th European Wireless Conference 2005 (2005).

[27] B. Xiao, B. Yu, C. Gao, Detection and Localization of Sybil Nodes in VANETs, in: Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks, DIWANS '06, 2006, pp. 1–8.

[28] M. S. Bouassida, G. Guette, M. Shawky, B. Ducourthial, Sybil Nodes Detection Based on Received Signal Strength Variations within VANET, I. J. Network Security 9 (2009) 22–33.

[29] S. Park, B. Aslam, D. Turgut, C. C. Zou, Defense against sybil attack in vehicular ad hoc network based on roadside unit support, in: Proceedings of the 28th IEEE Conference on Military Communications, MILCOM'09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 37–43. URL: `http://dl.acm.org/citation.cfm?id=1856821.1856828`.

[30] D. Cerri, A. Ghioni, Securing AODV: The A-SAODV Secure Routing Prototype, Comm. Mag. 46 (2008) 120–125.

[31] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561 (Experimental), 2003. URL: `http://www.ietf.org/rfc/rfc3561.txt`.

[32] V. Naumov, T. Gross, Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks, in: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, 2007, pp. 1919–1927.

[33] D. Johnson, A. Menezes, S. Vanstone, The Elliptic Curve Digital Signature Algorithm (ECDSA), International Journal of Information Security 1 (2001) 36–63. doi:`10.1007/s102070100002`.

[34] S. Berkovits, S. Chokhani, J. Furlong, J. Geiter, J. Guild, Public key infrastructure study: Final report, Technical Report, MITRE Corporation for NIST, 1995.

[35] C. Zhang, R. Lu, X. Lin, P.-H. Ho, X. Shen, An efficient identity-based batch verification scheme for vehicular sensor networks, in: INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, 2008, pp. 246 –250.

[36] C. N. Chong, Z. Peng, Secure audit logging with tamper-resistant hardware, in: 18th IFIP International Information Security Conference (IFIPSEC), volume 250 of IFIP Conference Proceedings, Kluwer Academic Publishers, 2003, pp. 73–84.

[37] A. Vargas, Objective modular network testbed in C++ (OMNET++), 2032. Version 4.2. Available: `www.omnetpp.org`.

[38] A. Ariza, INEMANET Framework for OMNeT++, `http://github.com/inetmanet/inetmanet/tree/master`, 2013.

[39] D. Krajzewicz, G. Hertkorn, C. Rössel, P. Wagner, SUMO (simulation of urban mobility); an open-source traffic simulation, in: 4th Middle East Symposium on Simulation and Modelling (MESM2002), MESM2002, 2002, pp. 183–187.

[40] V. Taliwal, D. Jiang, H. Mangold, C. Chen, R. Sengupta, Empirical determination of channel characteristics for DSRC vehicle-to-vehicle communication, in: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, VANET '04, ACM, New York, NY, USA, 2004, pp. 88–88.