BECSI: Bandwidth Efficient Certificate Status Information distribution mechanism for VANETs

Carlos Gañán *[1], Jose L. Muñoz[1], Oscar Esparza[1], Jonathan Loo[2], Jorge Mata-Díaz[1], and Juanjo Alins[1]

[1] *Telematics Department, Universitat Politècnica de Catalunya, Barcelona, Spain*
[2] *Computer Communications Department, Middlesex University, London, UK.*

Abstract

Certificate revocation is a challenging task, especially in mobile network environments such as vehicular ad Hoc networks (VANETs). According to the IEEE 1609.2 security standard for VANETs, public key infrastructure (PKI) will provide this functionality by means of certificate revocation lists (CRLs). When a certificate authority (CA) needs to revoke a certificate, it globally distributes CRLs. These lists must be distributed as quickly and efficiently as possible without over-burdening the network. In this article, we propose BECSI, a Bandwidth Efficient Certificate Status Information mechanism to efficiently distribute certificate status information (CSI) in VANETs. By means of Merkle hash trees (MHT), BECSI allows to retrieve authenticated CSI not only from the infrastructure but also from vehicles acting as mobile repositories. Since these MHTs are significantly smaller than the CRLs, BECSI reduces the load on the CSI repositories and improves the response time for the vehicles. Additionally, BECSI improves the freshness of the CSI by combining the use of delta-CRLs with MHTs. Thus, vehicles that have cached the most current CRL can download delta-CRLs to have a complete list of revoked certificates. Once a vehicle has the whole list of revoked certificates, it can act as mobile repository.

Keywords: PKI, Revocation, VANET.

1 Introduction

Vehicular ad-hoc networks (VANETs) have recently attracted extensive attentions as a promising technology for revolutionizing the transportation systems. VANETs consist of entities including On-Board Units (OBUs) and infrastructure Road-Side Units (RSUs). Mobile nodes are capable of communicating with

*Electronic address: carlos.ganan@entel.upc.edu; Corresponding author

each other (i.e. Vehicle to Vehicle Communication -V2V communication) and with the RSUs (i.e. Vehicle to Infrastructure Communication -V2I communication). Multi-hop communication facilitates information exchange among network nodes that are not in direct communication range [1, 2], by means of short range wireless technology based on IEEE 802.11p.

Obviously, any malicious behaviors, such as injecting beacons with false information, modifying and replaying the previously disseminated messages, could be fatal to the other users. Thus, identifying the message issuer is mandatory to reduce the risk of such attacks. According to the IEEE 1609.2 standard [3], 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 [3] 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.

As VANETs can have a great amount of nodes (i.e. vehicles), CRLs will be large. Moreover, each vehicle in the network will own many temporary certificates (also called pseudonyms) to protect the users' privacy. Consequently, these lists will require hundreds of Megabytes [4–6]. However, distributing and updating CRLs to all vehicles raises a challenge. If there are no more communication media than the own VANET, no trusted-third parties (like the corresponding CA) can be assumed to be permanently available. Thus, online certificate status protocol (OCSP) [7] or, in general, any online solution is not suitable for this context. Several CRLs distribution protocols have been proposed for this purpose. For instance, to distribute these lists efficiently, authors in [8] proposed revocation using compressed CRLs. They divided the CRL into several self-verifiable parts and strongly reduced its size by using Bloom filters. Authors in [5] also propose the use of Bloom filters to store the revoked certificates for increasing the search speed in the CRL. On the other hand, authors in [9] proposed to use regional CAs and short lived certificates to decrease the number of entries in the CRL. We provide more information about these and other similar proposals in Section 2 but as a general conclusion, we could say that most of the research efforts in this context have been put on trying to reduce the size of the CRL, either trying to split it or trying to compress it.

In this article, we address the CRL distribution problem by exploiting the combination of three well-known mechanisms: (1) delta-CRL [10], (2) Merkle hash tree (MHT) [11], and (3) one-way hash chain [12]. By combining these three mechanisms, we design a Bandwidth Efficient Certificate Status Information (BECSI) protocol, that allows increasing the availability and freshness of the certificate status information (CSI) and at the same time reduces the bandwidth necessary to check the validity of a given certificate. BECSI takes advantage of V2V communication to create mobile repositories so that vehicles do not have to rely solely in the RSUs to obtain CSI. We aim to improve the distribution of CSI by transmitting the revocation information that is unknown to a particular user

during the validity period of a CRL. The main idea behind BECSI is to allow vehicles requesting for new CSI during the validity period of the current CRL. Thus, revocations that occur during the validity period of the CRL will not be unknown to the vehicles during this whole validity period, reducing the risk of operating with an unknown revoked certificate. Therefore, BECSI reduces the peak bandwidth load associated with the CSI requests as there are more entities in the network that can answer these requests. To achieve, we combine the issuance of delta-CRLs and MHT.

By using the underlying concept of delta-CRLs, we implement a more efficient way of using of distributing CSI inside the VANET. To help minimize frequent downloads of lengthy CRLs, delta-CRLs are published aperiodically. On the other hand, BECSI codes the information included in the CRL and delta-CRLs in different MHTs. Using these MHTs, vehicles are able to act as mobile repositories. To achieve that, we embed some little extra information to the CRL such that allows us to create an efficient and secure request/response protocol. In more detail, we propose a way of efficiently embedding a MHT within the structure of the standard CRL to generate the so-called *extended-CRL* and *extended-delta-CRL*. To create these extended lists, we use an standard way of adding extra information to the CRL. Our extension contains all the necessary information to allow any vehicle or VANET infrastructure element that possesses the *extended-CRL* to build the BECSI tree, i.e., a hash tree with the CSI of the CRL. Using this BECSI tree, any entity possessing the *extended-CRL* can act as repository and efficiently answer to certificate status checking requests of other vehicles. As we will demonstrate by simulation, this makes the distribution of CSI more efficient than distributing complete CRLs (even they are compressed), reducing the data that have to be transmitted over the VANET. We must stress that any entity possessing an *extended-CRL* can act as BECSI repository but that a BECSI repository is not a TTP. In other words, BECSI is offline, which means that no online trusted entity (like a CA) is needed for authenticating the responses produced by BECSI repositories.

The rest of this paper is organized as follows. In Section 2, we present the background related to our mechanism. In Section 3 we describe in depth BECSI. In Section 4, we evaluate the proposed mechanisms. Finally, Section 5 concludes this paper.

2 Background

In this section, first we start describing existing revocation proposals for VANET. Then, we give a brief overview of Merkle Hash Trees (MHT) [11], which is one of the foundations of the proposed certificate validation mechanism. Finally we describe the basics of hash chains.

2.1 VANET revocation mechanisms

2.1.1 Centralized revocation approaches

The IEEE 1609.2 standard [3] 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.

Other proposals in the literature also assume the existence of a TTP to provide the revocation service. Raya *et al.* [13] propose the use of a tamper-proof device (TPD) to store the certificates. A TTP is in charge of preloading the cryptographic material in the TPD. 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. 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.

However, even compressed, timely distributing CRLs to all vehicles is not trivial. Some authors [9, 14], instead of using a single central authority, have proposed the use of regional certification authorities which must develop some trust relationships. Papadimitratos *et al.* [15] suggest restricting the scope of the CRL within a region. Visiting vehicles from other regions require to obtain temporary certificates. Thus, a vehicle will have to acquire temporary certificates if it is traveling outside its registered region. The authors also propose breaking the Certificate Revocation List (CRL) into different pieces, then transmitting these pieces using Fountain or Erasure codes, so that a vehicle can reconstruct the CRL after receiving a certain number of pieces. Similarly, in [16], each CA distributes the CRL to the RSUs in its domain through Ethernet. Then, the RSUs broadcast the new CRL to all the vehicles in that domain. In the case RSUs do not completely cover the domain of a CA, V2V communications are used to distribute the CRL to all the vehicles [17]. This mechanism is also used in [18, 19], where it is detailed a public key infrastructure mechanism based on bilinear mapping. Revocation is accomplished through the distribution of CRL that is stored by each user.

2.1.2 Decentralized revocation approaches

Decentralized revocation mechanisms provide the revocation service without assuming the existence of a TTP. Some proposals in the literature divert from the IEEE 1609.2 standard and use online status checking protocols instead of CRLs to provide a revocation service in a decentralized manner. This is the case, of the Ad-hoc Distributed OCSP for Trust (ADOPT) [20], which uses cached

OCSP responses that are distributed and stored on intermediate nodes. Other group of proposals bases the revocation service on detecting a vehicle to be misbehaving by a set of other vehicles. Then, the detecting set may cooperatively revoke the credential of the misbehaving node from their neighborhood. Moore *et al.* proposed in [21] a revocation mechanism aiming to prevent an attacker from falsely voting against legitimate nodes. Raya *et al.* in [13] proposed a mechanism to temporarily revoke an attacker if the CA is unavailable. To do so, the number of accusing neighbor users must exceed a threshold. A similar mechanism based also on vehicle voting is proposed in [22]. Again, by means of a voting scheme, a vehicle can be marked as misbehaving and then be revoked by its neighbors.

Another proposal uses a game-theoretic revocation approach to define the best strategy for each individual vehicle [23, 24]. These mechanisms provide incentives to guarantee the successful revocation of the malicious nodes. Moreover, thanks to the records of past behavior, the mechanism is able to dynamically adapt the parameters to nodes' reputations and establish the optimal Nash equilibrium on-the-fly, minimizing the cost of the revocation.

Finally, there are some hybrid approaches that are neither totally centralized nor decentralized ([25, 26]). For instance, authors in [27] propose the use of authenticated data structures to issue CSI. Using these schemes, the revocation service is decentralized to transmit the CSI but still depends on a CA to decide when a node should be evicted from the VANET.

2.2 The Merkle Hash Tree

A Merkle hash tree (MHT) [11] 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, as it is described in [28] for MANETs (Mobile Ad Hoc Networks). A sample MHT is presented in Fig. 1. This hash tree is binary because each node has at most two children or equivalently, two sibling nodes are combined to form a parent node in the next level. We will call these siblings as "left" and "right" and a detailed explanation of how to build the hash tree for BECSI is given in Section 3.3.

A MHT relies on the properties of the One Way Hash Functions (OWHF). It exploits the fact that an OWHF is at least 10,000 times faster to compute than a digital signature, so the majority of the cryptographic operations performed in the revocation system are hash functions instead of digital signatures. In addition, by storing the internal node values, it is possible to verify that any of the leaf nodes is part of the tree without revealing any of the other data.

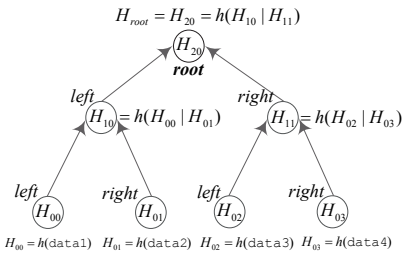

Figure 1: Sample binary Merkle Hash Tree.

2.3 Hash chains

The idea of "hash chain" was first proposed by Lamport [12] in 1981 and suggested to be used for safeguarding against password eavesdropping. A hash chain $\mathcal{C}$ is a set of values $s_0, ..., s_n$ for $n \in \mathbb{Z}$ such that $s_i = h(s_{i-1})$ for some one-way hash function h, where $i \in [1, n]$ and s_0 is a valid input for h.

Note that hash chains are preimage resistent, i.e., by knowing s_i, s_{i-1} cannot be generated by those who do not know the value s_0, however given s_{i-1}, its correctness can be verified by hashing $h(s_i)$. This property of hash chains has evolved from the property of one-way hash functions. Additionally, hash chains are also second preimage resistent, collision resistent and generate pseudo-random numbers.

In most of the hash-chain applications, first s_n is securely distributed and then the elements of the hash chain are spent (or used) one by one by starting from s_{i-1} and continuing until the value of s_0 is reached. At this point the hash chain is said to be exhausted and the whole process should be repeated again with a different s to reinitialize the systems.

3 BECSI: Bandwidth Efficient Certificate Status Information distribution mechanism

In this Section, we present BECSI, a bandwidth efficient mechanism for certificate status checking over VANETs based on the use of Merkle Hash Trees and hash chains. First we introduce the motivation, goal and security architecture needed to support BECSI, and next we describe the mechanism in depth.

3.1 Motivation and Goal

Despite the short-comings related to propagation of revocation information, the need for trusted authorities like CAs to ensure authentication has motivated researchers to propose PKI based security for vehicular networks. Mainly, these mechanisms intend to provide the following set of requirements:

1. *Reliability*: The revocation service must be available at all times.
2. *Memory*: Minimum amount of memory should be required as validation is often carried out in constrained environments.
3. *Bandwidth*: Communication bandwidth should be minimal.
4. *Freshness*: Revocation data should be as updated as possible.

Proposals described in Section 2 mainly deal with the bandwidth requirement. By compressing the CRL, using state-of-the-art coding techniques or partitioning the CRL, these approaches reduce the time required to download CSI. In addition, authors intend to provide a reliable revocation service by decentralizing the CSI distribution points. However, none of these works deals with the freshness of the CSI. With BECSI we aim not only to reduce the communication overhead but also to increase the availability and freshness of CSI while keeping a reasonable computation cost.

CRLs are normally published in intervals meaning that there will not be any new revocation information available between the issuance and the update of the CRL. Newer revocations will thus be delayed until the next update occurs. High-security applications (e.g. safety applications) cannot cope with this lack of fresh information and render the traditional CRL approach almost useless in VANETs. To solve these problems, BECSI includes an extension to the standard CRL that allow RSUs to act as an offline repositories. Thus vehicles do not have to download the whole extended CRL, and they can just query about the status of a particular certificate.

3.2 Security Architecture

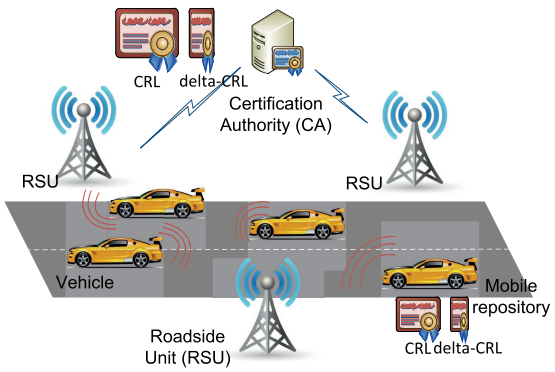

Figure 2: System Architecture.

The security architecture is an adaptation of a mesh PKI system to a vehicular scenario constructed of peer-to-peer CA relationships. This architecture consist of 4 different types of nodes (see Fig. 2):

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 certificates 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, so it should be assumed that it cannot be compromised by any attacker. In fact, in our proposal CAs are the only trusted entities within the network.

2. *Road-Side Units*: RSUs are fixed entities that are fully controlled by the CA. They can access the CA anytime because they are located in the infrastructure-side, which does not suffer from disconnections. If the CA considers that an RSU has been compromised, the CA can revoke it.

3. *Vehicles*: They are the clients of the network. They have their cryptographic material stored in a TPD. Vehicles can check the validity of a certificate using V2I or V2V.

4. *Mobile Repositories*: Mobile repositories are vehicles that have previously downloaded the CRL/delta-CRLs and are willing to response to cerficate status requests from other vehicles.

3.3 BECSI Tree

In this section, we introduce the data structure that BECSI uses to handle the revocation service. In this sense, we define the BECSI tree as a composite Merkle Hash Tree (see section 2). This tree consists of:

- A *base*-tree which is constructed using the serial number of the revoked certificates contained in the base-CRL.

- A set of Δ-trees which are constructed from the serial number of the certificates that are revoked during the validity interval of the base-CRL, i.e, they are constructed from the data contained in the delta-CRLs.

3.3.1 BECSI *base*-tree

The *base*-tree is a binary hash tree where each node represents a revoked certificate that is contained in the base-CRL. We denote by $N_{i,j}$ the nodes within the BECSI *base*-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 $N_{i,j}$ (see Fig. 3).

We denote by $N_{i,j}$ the nodes within the MHT where i and j represent respectively the i-th level and the j-th node. We denote by $H_{i,j}$ the cryptographic variable stored by node $N_{i,j}$.

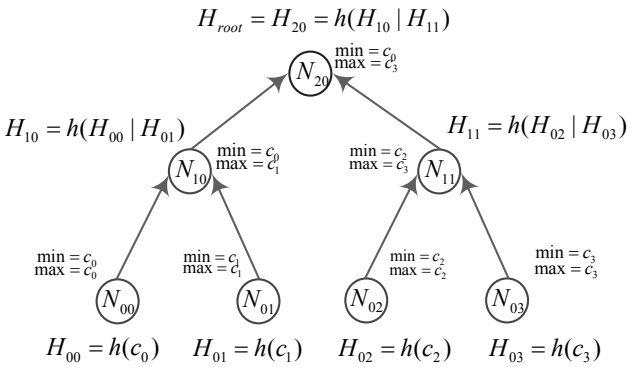

Figure 3: Sample BECSI *base*-tree.

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 have been revoked,

$$\Phi = \{c_0, c_1, \ldots, c_j, \ldots, c_n\}\,. \tag{1}$$

Where c_j is the data stored by leaf $N_{0,j}$. Then, $H_{0,j}$ is computed as:

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

where h is a OWHF.

To build the MHT, a set of t adjacent nodes at a given level i ($N_{i,j}$, $N_{i,j+1}$, $\ldots$, $N_{i,j+t-1}$) 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 t cryptographic variables:

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

At the top level there is only one node called the "root". H_{root} is a digest for all the data stored in the MHT.

The sample MHT of Fig. 1 is a binary tree because adjacent nodes are combined in pairs to form a node in the next level ($t = 2$) and $H_{root} = H_{2,0}$.

We define the the $\mathcal{D}$igest as the concatenation of the certification authority distinguished number, the root hash and the validity period of the certificate status data. Once created, the $\mathcal{D}$igest is signed by the CA.

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

We denote as the $\mathcal{P}\text{ath}_{c_j}$ as the set of cryptographic values necessary to compute H_{root} from the leaf c_j.

It is worth noting that $\mathcal{D}$igest is trusted data because it is signed by the CA and it is unique within the tree while $\mathcal{P}$ath is different for each leaf. Thus, If

the MHT provides a response with the proper $\mathcal{P}\text{ath}_{c_j}$ and the MHT $\mathcal{D}$igest, any vehicle can verify whether $c_j \in \Phi$.

For instance, let us suppose that a certain user wants to find out whether c_1 belongs to the sample MHT of Fig. 1. Then,

$\mathcal{P}\text{ath}_{c1} = \{H_{0,0}, H_{1,1}\}$,

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

The response verification consists in checking that $H_{2,0}$ computed from the $\mathcal{P}_{c_1}$ matches $H_{2,0}$ included in the $\mathcal{D}$igest:

$$H_{root} = H_{2,0} = h(h(h(c_1)|H_{0,0})|H_{1,1})\,.$$

Note that the BECSI *base*-tree can be built by a trusted third party (e.g. a CA) and distributed to a non-TTP because a leaf cannot be added or deleted to Φ without modifying H_{root}, which is included in the $\mathcal{D}$igest and as the $\mathcal{D}$igest is signed, it cannot be forged by a non-TTP. To do such a thing, an attacker would need to find a pre-image of a OWHF which is computationally infeasible by definition.

3.3.2 BECSI Δ-trees

BECSI Δ-trees are constructed in the same way that the *base*-tree. However they present two differences with respect to the *base*-tree:

- Each leaf of the Δ-trees refers to certificates that were revoked during the validity interval of the base-CRL.
- The root of the Δ-tree is calculated by hashing the top-hash of the tree with the corresponding value of a hash chain. For more details about the construction of the hash chain see Section 3.4.3.

Fig. 4 shows the simplest possible Δ-tree which contains only two revoked certificates.

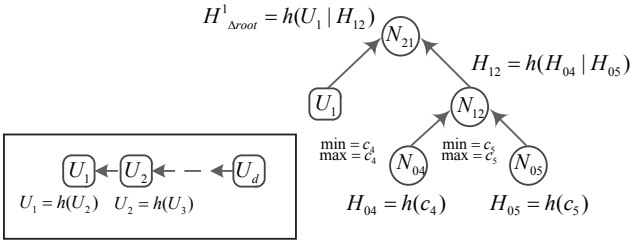

Figure 4: Sample BECSI Δ-tree.

Note that these Δ-trees have the same properties that the *base*-tree, so that the root node is unique and cannot be forged. Thus, the $\mathcal{D}$igest is composed as:

$$\mathcal{D}\text{igest}_{\Delta_i} = \{DN_{CA}, H^i_{\Delta root}, ValidityPeriod\}_{SIG_{CA}}\,.$$

Similarly, the $\mathcal{P}$ath consist of the set of cryptographic values necessary to compute $H^i_{\Delta root}$ from the leaf c_j. Note that this path is shorter in the case of the leafs of the Δ-trees because they contain less revoked certificates than the *base*-tree. The length of the Δ-trees is fixed as they are constructed from the delta-CRLs that have fixed size.

3.4 Operating Mode

BECSI consists in four phases. During the first phase of *Bootstraping*, the CA creates the "extended-CRL", that is, a CRL in which a signed extension is appended. This extension will allow non-trusted third parties (non-TTP) to answer CSI requests in an off-line way when required. Once this extended-CRL has been constructed, it is distributed to the RSUs. In the second phase of *Repository Creation*, a non-trusted entity (i.e. a RSU or a vehicle) gets the extended-CRL and becomes a CSI repository for other VANET entities. Next, during the third phase *CSI Update*, the CA creates "extended-delta-CRLs" of fixed size. A delta-CRL is a time-stamped digitally signed revocation list containing information about new revocations that occurred since the issuance of a prior base-CRL[1]. To construct these fixed-size delta-CRLs, the CA has to wait to have enough new revoked certificates. Therefore, the issuance of these delta-CRLs is aperiodic. Once, the delta-CRL is constructed, the CA appends a signed extension corresponding to the root node of the Δ-tree. The extended-delta-CRLs are distributed to the RSUs and mobile repositories. Moreover, the CA broadcasts the number of issued extended-delta-CRLs to avoid CSI suppression attacks. Finally, in the fourth stage of *Certificate Status Checking*, vehicles can use an efficient protocol to obtain the CSI from any available VANET repository. Henceforward, we give a more detailed description of these three stages.

3.4.1 Bootstraping

In this first phase, 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-TTP (e.g. RSUs and vehicles inside the VANET) to act as repositories and answer to CSI requests. All the tasks of this system initialization are performed in the CA locally.

These are the steps that the CA must carry out:

1. The CA creates a *tbs-CRL* (to be signed CRL), i.e., a list including the serial number of the certificates that have been revoked (along with the date of revocation), the identity of the CA, some time-stamps to establish the validity period, etc.

2. The CA creates the BECSI *base*-tree, i.e., a MHT constructed with the serial numbers within the previous *tbs-CRL* as leaves of the tree. The BECSI

[1] A base-CRL is a complete CRL that contains a complete list of revoked certificates, to which the revocation list in the delta-CRL needs to be applied to produce the latest list of revoked certificates. Base and delta CRLs have similar data structures.

base-tree is a binary tree, and is constructed following the methodology explained in Section 3.3. The leaves of the *base*-tree are ordered by increasing serial number. Therefore, the bottom left leaf stores the revoked certificate with lowest serial number. Note that if the BECSI base-tree is formed by an odd number n of leaves, there is a leaf $N_{0,n-1}$ that does not have a pair. Then the single node is simply carried forward to the upper level by hashing its $H_{0,n-1}$ value. We proceed in the same way if any i-th level is formed by an odd number n of nodes. Once created the MHT, the CA obtains the root hash.

3. The CA calculates the extension, which consists basically of the $\mathcal{D}$igest and the first value U_0 of a hash chain. The hash chain will be used to make users aware of the number of issued delta-CRLs. Recall that the $\mathcal{D}$igest is calculated as the concatenation of the CA distinguished number, the root hash of the base-tree and the validity period of the CSI, and after that signed by the CA. Obviously, the distinguished number and the validity period should be the same than the ones contained in the tbs-CRL. In fact, the BECSI base-tree is just a different way of representing the CSI, but the hash tree will be valid during the same time and will provide the same information than the CRL. Once calculated, this $\mathcal{D}$igest is appended to the tbs-CRL.

4. The CA creates the hash chain. To that end, the CA picks a random value for U_d. By hashing this value iteratively, the CA forms a one-way chain of self-authenticating values, and assigns the values sequentially to the time intervals (one value per delta-CRL). The last value of the chain U_0 is appended to the tbs-CRL along with the $\mathcal{D}$igest, generating the *tbs-extended-CRL*.

5. The CA signs the tbs-extended-CRL, generating the *extended-CRL*. Notice that this second overall signature not only authenticates all the CSI, but also binds this CSI to the $\mathcal{D}$igest. The *extended-CRL* is only slightly larger than the standard CRL, as we will show later in Section 4.

6. Finally, the CA distributes copies of the *extended-CRL* to the designated RSUs, which will act as the typical PKI repositories, in the same manner as they would do with a standard CRL.

After this first phase, 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-TTP in possession of it can generate again the BECSI base-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 BECSI base-tree and can answer to CSI requests in an off-line way.

3.4.2 CSI Repositories creation

In this phase, RSUs and freewill vehicles become new CSI repositories of the VANET. Vehicles that become mobile repositories allow the distribution of CSI in areas with poor coverage. To become a repository an entity must follow the following steps:

1. The entity obtains the *extended-CRL* (and *extended-delta-CRLs*)either from the CA or from another entity that has an up-to-date copy of the *extended-CRL* (and *extended-delta-CRLs*) 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* (and *extended-delta-CRLs*) has been downloaded, the entity verifies that the signature of the *extended-CRL* (and *extended-delta-CRLs*) is valid and corresponds to the CA. If so, the entity generates locally the BECSI base-tree (and Δ-trees) using the serial numbers within the *extended-CRL* (and *extended-delta-CRLs*) 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* (and *extended-delta-CRLs*) entries must match the signed root value contained in the $\mathcal{D}\text{igest}_{base}$ (and $\mathcal{D}\text{igest}_{\Delta_i}$).

3. At this moment, the entity can respond to any status checking request from any vehicle until the corresponding $\mathcal{D}$igest expires.

3.4.3 CSI Update

After the first two phases, any entity of the VANET is capable of downloading the CRL from a repository or it can just check the status of a given certificate using the capabilities of a MHT. However, in order to improve the freshness of the revocation information and avoid potential bottlenecks when obtaining new CSI, BECSI also provides CSI updates during the validity interval of the CRL.

To alleviate high CRL distribution costs, BECSI uses a hybrid delta-CRL scheme. BECSI issues a variable number of delta-CRLs during the validity interval of the base CRLs (as shown in Fig. 5), reducing the total bandwidth load on the CRL distribution points (RSU and mobile repositories). The size of these delta-CRLs is fixed a priori by the CA. Consequently, the number of delta-CRLs issued during the validity interval of the base-CRL depends on the number of revoked certificates during this interval.

To ensure that any vehicle entity is aware of the number of issued delta-CRLs, the CA discloses a value U_i of a hash chain each time a new delta-CRL is issued. This hash value allows users to make sure of how many Δ-trees have been published by the CA. Thus a non-TTP cannot lie about the amount of revocation information that has been published. Note that the corresponding value U_i is used to calculate the root value of the Δ-tree, binding the Δ-tree to the U_i. BECSI takes advantage of the physical layer used in VANETs to transmit the hash value U_i to vehicles.

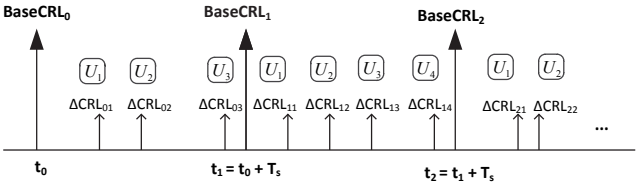

Figure 5: Delta-CRLs Issuance Scheduling.

The physical layer in VANETs is based on the Dedicated short range communication (DSRC) protocol [2]. DSRC is a 75 MHz band in the 5.9 GHz frequency range with seven non-overlapping channels. Two different channel types are described for use in DSRC. The first type is the control channel, referred to as CCH, which is a single channel reserved for short, high-priority application and system control messages [3]. During the CCH, every node broadcasts a beacon that provides trajectory and other information about the vehicle. The other type of channel is the service channel, or SCH, which has six different 10 MHz channels that support a wider range of applications and data transfer. During CCH time channel activities on SCH are suspended and vice versa. BECSI uses the CCH to transmit the corresponding U_i. Each node in the VANET monitors the CCH during time periods designated as control channel intervals. The time period for an entire CCH Interval and SCH Interval is called a Sync Interval (see Fig. 6). Between CCH intervals, nodes may switch to participate on a SCH for applications such as file downloads.

Figure 6: CCH/SCH timing.

Each regional CA sends to RSUs an authenticated message M containing the corresponding U_i and a time-stamp.

$$CA \rightarrow RSUs : M = [U_i, TimeStamp]_{Sign_{CA}}$$

Note that regional CAs are expected to have a wireline to communicate with their corresponding RSUs. The time stamp included in the message allows vehicles to verify the freshness of the message. Thus, it is avoided potential forgery or replay attacks. The size of this message is 72 bytes:

- 64 bytes for the ECDSA-256 CA's signature.
- 4 bytes for the timestamp representing seconds UTC since the epoch ('1970-01-01 00:00:00' UTC).
- 4 bytes for representing the U_i value.

During the CCH interval, RSUs broadcast this message to OBUs in range. However, not in every CCH interval M is sent. Depending on the certificate revocation rate, each regional CA will choose the rate at which they have to transmit the U_i to the vehicles. Normally, vehicles will remain under the coverage of an RSU for more than $100ms$. Therefore, CAs have to adjust the frequency at which M is sent to avoid vehicles receiving multiple copies of the same message. Notice that as M is signed by the CA, any vehicle can act as repository and transmit this message without being able to modify it. The hash chain is initialized with secret nonce that the CA generates (U_d) and includes in the *extended*-CRL. By hashing U_d, the other U_i nodes of the chain are calculated. As the validity interval of the CRL is finite, the length of the chain is also finite, i.e., U_0 is the last node of the chain that is calculated after hashing d times U_d. Thus each value can be calculated applying a hash function h to the previous value, and the first value of the hash chain is the secret nonce U_d.

$$U_d \xrightarrow{h} U_{d-1} \xrightarrow{h} \ldots \xrightarrow{h} U_i \xrightarrow{h} U_2 \xrightarrow{h} U_1 \xrightarrow{h} U_0$$

On the other hand, BECSI not only issues delta-CRLs aperiodically, it also includes an extension in the delta-CRLs as it does with the base-CRL. Thus, any VANET entity that has cached the extended-delta-CRL can construct the BECSI Δ-tree (as shown in Sec.2.2). With this MHT, a non-TTP can respond to any entity requesting the status of a particular certificate. The response can be authenticated by the requesting party by means of the extension.

3.4.4 Certificate status checking

After the third phase, RSUs and some vehicles 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 validate the status of a certificate. Under BECSI, vehicles have to option to check the status of a certificate:

- Downloading the standard CRL and the available delta-CRLs from any repository. This option is desirable when the connectivity to the infrastructure is high and the network congestion is low. For instance, in a urban scenario during non-rush hours where the deployed infrastructure should be enough to serve the CSI.

- Requesting the status of a particular certificate to any repository. With this option, the requesting vehicle only gets the status of a single certificate, so the bandwidth load is low. Vehicles should use this option when they need a quick response (e.g. authenticating safety messages), or when the VANET conditions are not good enough to download the whole CRL and delta-CRLs.

Independently of the option, vehicles that need to check the status of a certificate must locate a valid repository. To do so, vehicles use a Service Discovery Protocol (SDP) to find a RSU or a vehicle that is acting as repository. Once the

repository has been located, the vehicles can query for the CRL or query for the status of a particular certificate using the following status checking protocol.

The protocol for status information exchange is based on the hash tree structure and it allows checking the integrity of a single *extended-CRL* or *extended-delta-CRL* entry with only some hash material plus the $\mathcal{D}$igest (included in the extension) and the corresponding U_i. On the one hand, this is much more efficient than broadcasting the entire *extended-CRL* and the *extended-delta-CRLs*. 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.

Hence, a vehicle that needs to check the status of a certificate must follow the next steps:

1. The vehicle uses a service discovery protocol to find either a RSU or a mobile repository inside its coverage range for status checking.

2. The vehicle sends the serial number of the certificate that is going to be verified to the repository. The repository searches the target certificate in the base-tree and the Δ-trees. In the case the certificate is found, the repository sends the $\mathcal{P}$ath, i.e., the hash values of the nodes of the base-tree (or Δ-tree) which are needed to calculate the signed root. To calculate the path, the repository follows a recursive algorithm that starts from the root and goes across the MHT until the target leaf is reached (see Algorithm 1).

Algorithm 1: Algorithm to calculate the $\mathcal{P}$ath of a given certificate.

```
Input: SN_target
Output: Path
foreach base-tree and Δ-tree_i do
    if SN_target ∈ Δ-tree_i then
        k = i
    else
        k = 0
N_ij = root_k;
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,k
```

3. The vehicle verifies that the H_{root} (or the H^i_{root}) calculated from the

$\mathcal{P}$ath matches the H_{root} (or the H^i_{root}) contained in the $\mathcal{D}$igest$_{base}$ (or the $\mathcal{D}$igest$_{\Delta_i}$).

Notice that as all H_{root} and all $\mathcal{D}$igest are signed by the CA, it is just as impractical to create falsified values of the $\mathcal{P}$ath as it is to break a strong hash function. In case the certificate is not revoked, the repository sends the adjacent leaves to the requested certificate. To this respect, the repository has to prove that a certain certificate (SN_{target}) does not belong to the set of revoked certificates (Φ). To prove that $SN_{target} \notin \Phi$, as the leaves are ordered, it is enough to demonstrate the existence of two leaves, a minor adjacent (SN_{minor}) and a major adjacent (SN_{major}) for the base-tree and each Δ-tree that fulfill:

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

So in the worst case, where d delta-CRLs have been published, the repository will have to send $2d+1$ $\mathcal{P}$aths to proof that a certificate is not revoked, i.e, the serial number is not contained neither in the base-CRL nor in the delta-CRLs. Note that in any case, the amount of data necessary to proof that is smaller than the whole CRL. Therefore, checking vehicles have to know exactly the number of published Δ-trees, i.e., to check that a certificate is not revoked it must corroborate that it does not belong to any MHT. In any case, the data that the repository needs to send to a node to perform the status checking can be placed in a single UDP datagram using 802.11p link-layer.

4 Performance Evaluation

In this section, we evaluate the efficiency of the proposed status checking protocol and we compare it with other certificates status management protocols designed for VANET. First, we define a set of metrics to compare the performance of revocation schemes. Then, BECSI is evaluated through simulation using NCTUns [29]. NCTUns was chosen for its advanced IEEE 802.11 model library and ability to integrate with any Linux networking tools.

4.1 Comparison Criteria

- *Query Cost (Q_{cost})*: This criterion measures the cost of certificate validity checking. The cost represents bandwidth requirement from repositories to vehicles. Therefore, we calculate this cost as the size of a CSI query (s_q) plus the size of its response (s_r):

$$Q_{cost} = s_q + s_r. \tag{4}$$

- *Request Ratio*: This metric captures the amount of requests that the VANET entities perform to update the CSI. If client validation requests arrive independent of each other, an exponential inter-arrival probability density function can be used to derive the request rate (R) for downloaded CRLs as in [30]:

$$R_t = N_{veh}\lambda e^{\lambda t}, \tag{5}$$

where N_{veh} is the total number of vehicles in the VANET and λ is the ratio of certificates validated per day by each vehicle.

- *Window of vulnerability (WOV)*: This criterion captures the risk of operating with cached CSI. It indicates how long the new revocation data might be held by CAs before being distributed to vehicles. In this paper, WOV is measured in number of hours, which is reasonable because typically CRLs are normally updated every day. We estimate the WOV not only taking into account the validity interval of issued CSI but also the ratio of unknown revoked certificates during this interval as in [31]. Thus,

$$WOV(t) = \frac{\rho(t - t_0)}{(1-\rho)T_c + \rho(t - t_0)}, \tag{6}$$

where T_c is the mean certificate lifetime, ρ is the revocation ratio of revoked certificates, and t_0 is the issuing instant of the CSI.

- *Scalability*: This criterion shows how a revocation mechanism scales in large VANETs, measured as the ratio of increased costs (in terms of update and query costs) over increased size of the vehicles (measured in the number of certificates, queries and revoked certificates). If we assume a stable certificate revocation rate and query rate, a larger VANET typically indicates more revoked certificates and queries in unit time.

4.2 Analytical Evaluation

In this section, we compare analytically the performance of BECSI to other certificate status validation mechanisms. To that end, we compare BECSI with these mechanisms in terms of aforementioned metrics.

4.2.1 Query Cost Analysis

First of all, we start estimating the size of a CRL in a vehicular environment.

Fig. 7 describes the size of each of the elements that compose a CRL. Note that, in a VANET, the size of the CRL will depend mainly on the number of revoked certificates, so that the size of the CRL header is negligible compared to the total size of the CRL. Let N_{veh} be the total number of vehicles in the region that the CRL needs to cover, ρ the average percentage of certificates revoked, L_f the lifetime of a certificate, and $\bar{s}$ the mean number of pseudonyms of a

CRL Header ($\sim$ 50 bytes)

- Issuer's name: 32 bytes (if X.500 name used)
- CRL issuance time (thisUpdate): 6 bytes
- Next CRL issuance time (nextUpdate): 6 bytes

List of revoked certificates (9 bytes per revoked certificate)

- Serial number : 3 bytes
- Revocation date : 6 bytes
- CRL entry extensions (e.g. revocation reason)

CRL general extensions (e.g. CRL Number)

Signature of CRL issuer (64 bytes for ECDSA-256 bit)

Figure 7: Key elements of X.509 v2 CRL

vehicle. Additionally, let N_{rev} be the number of non-expired certificates that were revoked, i.e., the number of certificates that the CRL contains. According to [32], the probability of a certificate being revoked follows an exponential distribution. Then, the probability of a given certificate to become revoked at any time period of its lifetime $i \in [0, \ldots, L_f]$ can be expressed as:

$$P_{rev}(i) = L_f e^{-i \cdot L_f}.$$

When a certificate is revoked at time period i of its lifetime, it stays in the CRL for $L_f - i$ time periods. Thus, the expected time a revoked certificate stays in the CRL can be estimated as:

$$E(L_f - i) = E(L_f) - E(i) = L_f - \frac{1}{L_f} = \frac{L_f^2 - 1}{L_f} \simeq L_f.$$

Then, we can estimate the mean number of revoked certificates in a CRL as:

$$\overline{N_{rev}} = N_{veh} \cdot \rho \cdot \bar{s} \cdot L_f.$$

Finally, we estimate the size of a CRL in a VANET. As shown in Fig.. 7, CRL entries will have varying sizes, but according to 1609.2 standard [3], 14 bytes per entry is a realistic figure, i.e, $s_e = 14$ bytes. The size of the CRL header is negligible compared to the total size of the CRL. According to NIST statistics [33], 10% of the certificates need to be revoked during a year, i.e., $\rho = 0.1$. Recall that in a VANET, each vehicle owes not only an identity certificate, but also several pseudonyms. The number of pseudonyms may vary depending on the degree of privacy and anonymity that it must be guaranteed. According to Raya, Papadimitratos, and Hubaux in [8] the OBU must store enough pseudonyms to

change pseudonyms about every minute while driving. This equates to about 43,800 pseudonyms per year for an average of two hours of driving per day. Haas, Hu, and Laberteaux in [34] recommend changing pseudonyms every 10 minutes, and driving 15 hours per week. This equates to 4,660 pseudonyms per year, but they recommend storing five years of pseudonyms for a total of about 25,000 pseudonyms per OBU. Therefore, we set $\bar{s} = 25,000$. Regarding to the certificate lifetime, according to [32], it ranges from 26 to 37 days. In this manner, we set the lifetime to 1 month. Therefore, the expected CRL size is:

$$CRL_{size} = \overline{N_{rev}} \cdot s_e = N_{veh} \cdot \rho \cdot \bar{s} \cdot L_f \cdot s_e$$

Assuming that a regional certification authority manages a very short population of around 50,000 vehicles, the expected CRL size is $CRL_{size} \simeq 145$ Mbytes.

On the other hand, the response size of BECSI (when using the MHTs to generate authenticated responses) is much smaller than a CRL as it consists only of the $\mathcal{D}$igest and the $\mathcal{P}$ath for a given certificate. Using the SHA-1 algorithm (hash size of 160 bits), and ECDSA-256 the size of the response of BECSI for 10,000,000 revoked certificates (including pseudonyms) is of approximately 725 bytes.

Mechanism	**Request size**	**Response size**	Query Cost
CRL	73 bytes	145 Mbytes	~145 Mbytes
Compressed CRL (Bloom Filter-2% false positives)	73 bytes	10 Mbytes	~10 Mbytes
ADOPT	66 bytes	586 bytes	652 bytes
BECSI tree	73 bytes	725 bytes	778 bytes

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

In terms of the total overhead introduced to the network, Table 1 shows the *Query Cost* for current proposed certificate validation mechanisms. Note that the request size is very similar for all the mechanisms. However, the size of the response varies significantly, e.g., BECSI and ADOPT response sizes are six orders of magnitude smaller than conventional CRL. Fig. 8 shows the size of the response for CRL, Compressed CRL, ADOPT [20] and BECSI depending on the number of revoked vehicles in the network. While ADOPT response size is constant, the size of the response when using CRL or a compressed version of the CRL increments with the number of revoked certificates. Notice that, the CRL size grows linearly with the number of revoked certificates, while BECSI response sizes describe a logarithmic growth. Therefore, in terms of Query Cost, BECSI is more efficient than CRL and the compressed CRL. Regarding ADOPT, its response size is slightly smaller than in BECSI, but it lacks of the benefits that BECSI provides to operate during disconnections. ADOPT relies on the fact that any vehicle stores the previously received CSI responses. In

vehicular scenarios the number of cached responses could be huge, and therefore, also a huge storage capacity is required in the vehicle. In addition, ADOPT does not guarantee that a vehicle obtains the status of a given certificate when needed. So, ADOPT has smaller responses, but it does not provide as fresh information as BECSI, it forces VANET nodes to store a large amount of CSI data and finally, it makes the network more vulnerable due to the potential unavailability of required CSI.

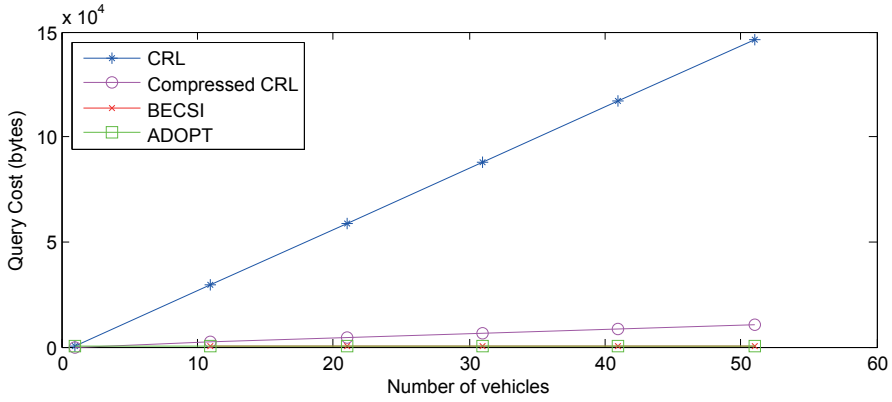

Figure 8: Response size vs. number of vehicles.

4.2.2 Request Ratio Analysis

In the traditional method of certificate revocation, each CRL includes a *nextUpdate* field that specifies the time at which the next CRL will be issued. Thus, once a relying party has obtained a CRL in order to perform a validation, it will not need to request any further information from the repository to perform future validations until the time specified in the *nextUpdate* field of the CRL in its cache has been reached. So, during the period of time in which a CRL is valid (i.e., the most current), each relying party will make at most one request to the repository for revocation information. This request will be made the first time after the current CRL is issued that the relying party performs a validation. Thus, the request ratio of the CRL decreases during the validity interval of the CRL following an exponential function (see. Fig. 9). Figure 9 shows the request rate for a CRL, issued using the traditional method, over the course of 24 hours. The graph in this figure was drawn assuming that a CRL was issued at time 0 and that no other CRLs were issued during the period of time shown in the graph. It was also assumed that there are 50,000 vehicles each validating an average of 10 certificates per day.

Figure 9 shows also an example of delta-CRLs issued in the traditional manner. In this example, vehicles download base CRLs at most once every 24 hours. Delta-CRLs are then obtained to ensure that validations are based on certificate

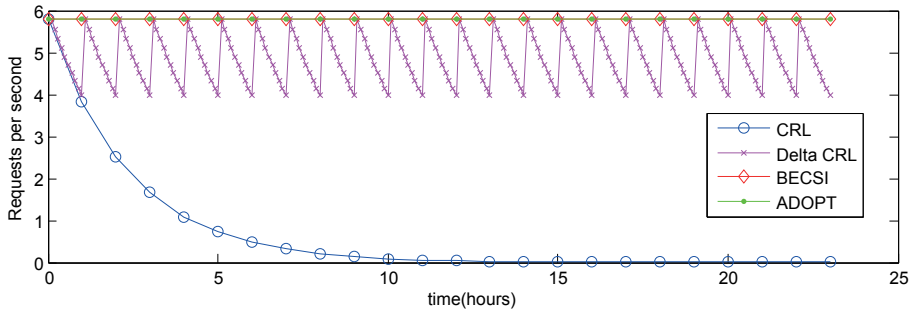

Figure 9: Request rate for different revocation mechanisms.

status information that is at most 1 hour old. Each validation will require access to a delta-CRL and its corresponding base CRL (either downloaded from the repository or generated locally from a delta-CRL and a previous base CRL). So, the request rate for delta-CRLs will be the same as the request rate for full CRLs in a system that does not use delta-CRLs. Base CRLs, on the other hand, will be downloaded less frequently.

Finally, regarding the cases of BECSI and ADOPT, the request rate is almost constant, i.e., every time a vehicle needs to check the status of a particular certificate they must query a repository. Note that this rate decreases with time, as vehicles also have the ability to store previously queried CSI. However, as the number of valid certificates is so large in VANETs, this decrement is imperceptible.

4.2.3 WOV Analysis

The window of vulnerability (WOV) affects update and query bandwidth requirements and/or repositories processing loads directly, while these two factors are two major features determining scalability of CSI issuing mechanisms. WOV presents a direct tradeoff between the security/ timeliness and system scalability. No window of vulnerability means high security and is thus desirable; however, it requires either timely certificate status update from CAs that can force a high update cost and incur security risk.

The traditional way of issuing CRL is the worst mechanism in terms of WOV. During the whole validity of the CRL, vehicles are unaware of new revoked certificates. Therefore, the WOV will increase during the validity of the CRL as there will be more unknown revoked certificates as times goes by. Figure 10 shows the WOV for a CRL issued periodically each 24 hours, and with a constant revocation rate $\rho = 0.1$. Note that the revocation rate determines the slope of the function, i.e., higher revocation rate will give higher WOV.

Compressed CRLs have the same WOV that the standard CRL as they are just a compressed version of the CRL issued with the same lifetime. In the same way, ADOPT also has the same WOV that a CRL. ADOPT presents a

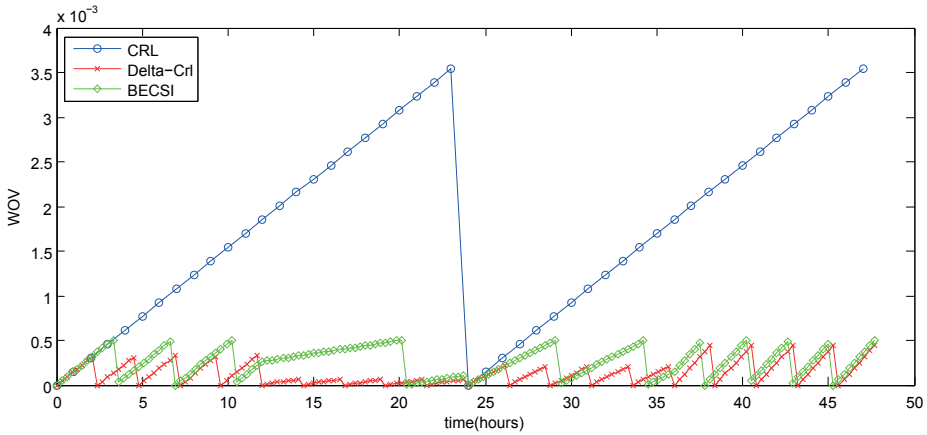

Figure 10: WOV for different revocation mechanisms.

distributed mechanism that takes advantages of V2V communications to issue cached CSI. However, the CSI source of this cached revocation information is a CRL. Therefore, the validity period of the cached information in ADOPT is the same that the validity period of the source CRL, i.e., the WOV is the same.

Figure 10 also shows the WOV for BECSI and delta-CRLs. Note that both mechanisms improve the WOV. Traditional delta-CRLs reduce the WOV as they are issued periodically during the validity of the base-CRL. Therefore, the interval of vulnerability of the base-CRL is reduced as many times as delta-CRLs are issued during the lifetime of the base-CRL. In the example shown in the figure, for each base-CRL issued each day, a delta-CRL is issued each 2.4 hours. Thus, 10 delta-CRLs are issued during the validity interval of the base-CRL, reducing the WOV ten times.

In the same way, BECSI also uses delta-CRLs to construct the tree structure. Therefore, the tree structures used in such scheme reduces not only update or query costs, but also the WOV. Recall that BECSI does not issue delta-CRLs periodically, but these are issued with a fixed size. In this sense, despite that fact that the WOV could be higher than with the traditional delta-CRL issuing mechanism, the maximum WOV is always constant. Thus CAs can manage the WOV by selecting the size of the delta-CRLs. In the example shown in the figure 10, the number of delta-CRLs issued during the validity interval of the base-CRL is reduced compared to the traditional delta-CRL issuing mechanisms. Note that in this example, BECSI's WOV is never higher than 0.0005.

4.2.4 Scalability Analysis

When the vehicular population is large, CRLs tend to become large imposing high bandwidth costs on the CRL distribution points. Hence traditional CRL-based schemes do not scale well. If clients have limited bandwidth capability as

is the case of the 802.11p, downloading large CRLs will be user-unfriendly.

With traditional delta-CRLs, the base-CRLs are issued less frequently (as shown in Fig. 5), this reducing the total bandwidth load on the CRL distribution points. However, that use of the traditional delta-CRL does not lead to a significant reduction in bandwidth as one would expect. If delta-CRLs are issued very frequently, there is no advantage in using traditional delta-CRLs. Therefore, although the scalability improves compared to simple CRLs mechanisms, traditional delta-CRLs scalability depends on the issuing periodicity of the delta-CRLs.

BECSI takes advantage of the delta-CRLs and optimize the issuing interval so that delta-CRLs remain constant in size. With BECSI, delta-CRLs always have the same size, but they are issued aperiodically. Thus, BECSI becomes more scalable than traditional delta-CRL where depending on the revocation rate the issuing period of the delta-CRLs could be bandwidth-inefficient. Moreover, BECSI also takes advantage of the capabilities of the V2V communication, allowing any vehicle in the network to become a mobile repository. In this sense, BECSI (as ADOPT), multiplies the number of potential repositories, and, therefore, its scalability is also increased.

4.3 Simulation

In the previous section, we have seen analytically that BECSI mechanisms outperform CRL in terms of Query Cost, WOV and scalability. Moreover, BECSI also improves other revocation mechanisms such as ADOPT when analyzing the availability of fresh CSI. In this section, we evaluate the proposed mechanism in a VANET scenario taking into account the specific characteristics of these networks. Using the simulator NCTUns [29], BECSI is evaluated.

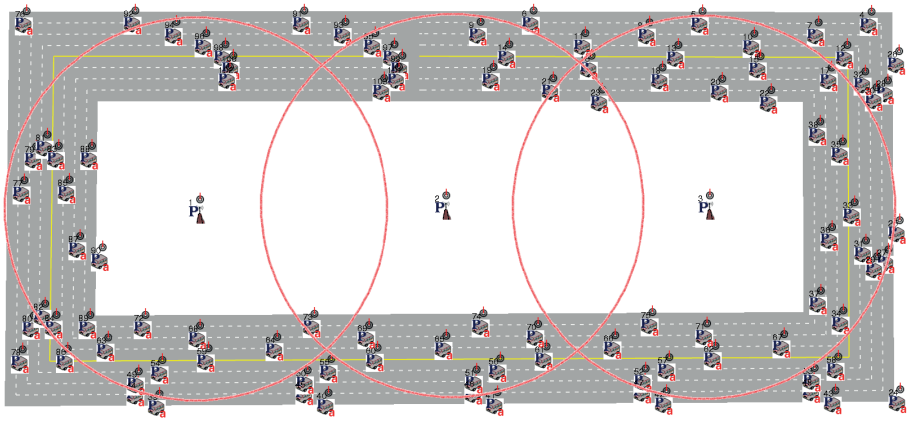

Figure 11: Simulation scenario.

The reference scenario is shown in Figure 11. This scenario consists of 4

two-lane roads forming a 1000x500m rectangle. Three RSUs are placed every 300 meters. Note that there are some areas of the highway that are not covered.

Table 2 summarizes the values of the configuration parameters used in the reference scenario. 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 [35].

Parameter	**Value**
Area	1000x500m
Number of RSUs	3
Number of OBUs	100
RSU Transmission range	300m
MAC	IEEE 802.11p
Propagation model	Nakagami
Number of caching nodes	20
Maximum speed	120 km/h

Table 2: Parameter values for the reference scenario.

Using this scenario as reference, firstly, we compare the CSI validation delay of the BECSI scheme with that of the classical PKI [3] under a well-deployed VANET. In the conducted simulation, we consider the cryptography delay only due to hashing operations and point multiplication operations on an elliptic curve, as they are the most time-consuming operations in the proposed protocol. Let T_{hash} and T_{mul} denote the time required to perform a pairing operation and a point multiplication, respectively, respectively. Elliptic curve digital signature algorithm is the digital signature method chosen by the VANET standard IEEE1609.2, where a certificate and signature verification takes $4T_{mul}$, and a signature generation takes T_{mul}. To verify a credential in the basic scheme described in Section 3.4.4, a vehicles must perform a hash operation to compute the current contents of leaf node in the BECSI-tree corresponding to SN_i. Finally, it performs $logN$ hash operations to compute the root of the BECSI-tree using the $\mathcal{P}$ath. Therefore, the total computation overhead when checking the status of a certificate is $T_{hash}(logN + 1) + 4T_{mul}$. In [36], T_{mul} are found for an MNT curve with embedding degree $k = 6$ that is equal to 0.6 ms. In our simulation, we use an Intel Core i7 950 (at 3.07GHz) which is able to perform 1015952 SHA-1 Hashes per second, i.e, $T_{hash} = 0,98\mu$s. Therfore the expected time to check the validity of a $\mathcal{P}$ath in BECSI with is 2.4 ms.

In VANETs, the most important issue in any revocation method is the delay of delivering the CSI to the vehicles to prevent that misbehaving vehicles from jeopardizing the safety of its neighbors. Consequently, we measure the revocation delay as delay from the moment a vehicle issues a CSI request until the moment the new CSI is received. Table 3 shows the average time spent by a vehicle to retrieve CSI from a repository.

It is worth noting that the worst mechanisms in terms of delay are the

Revocation Mechanism	Average Time	Standard Deviation
CRL (300 KB)	2,23 min	0,51 min
Compressed-CRL (20 KB)	7,01 sec	1,12 sec
Traditional Delta CRL (2.5-15 KB)	4,47 sec	2,12 sec
ADOPT (652 B)	705,06 ms	200,81 ms
BECSI Delta CRL (8 KB)	6,02 sec	0,05 sec
BECSI MHT (778 B-912 B)	483,02 ms	20,31 ms

Table 3: Time required to retrieve CSI.

traditional CRL and delta-CRL as requesting entities are downloading all the available CSI. However, the delay of the conventional CRL compared with the proposed BECSI protocol decreases with the number of CSI requests. The variations in time to download the CRL are due to the number of intermediate RSUs existing in the connection between the CA and the vehicle sending the revocation request. The average time to validate the status of a certificate in ADOPT is lower than BECSI because of the number of hops that are necessary to retrieve the cached CSI. BECSI in its MHT mode of operation is the fastest in average when validating the status of a certificate. However, this mode of operation has a also a notable deviation. While in ADOPT the high deviation is due to the number of hops, in BECSI this deviation is mainly due to the number of Δ-trees that a vehicle has to check when a certificate is not revoked. Note also, that there are also some deviations from the theoretical expected results. This is due to several reasons such as the non-uniform distribution of the mobile repositories, the distance to the repositories or the congestion of the channel. Figure 12 shows the number of vehicles that are able to download the CSI in a particular range time depending on the revocation mechanisms. As expected, with BECSI and ADOPT almost all the 100 vehicles are able to download and process the CSI in less than 1,5 seconds. However, with Delta-CRLs and compressed-CRLs it takes from 4 to 8 seconds to retrieve the CSI.

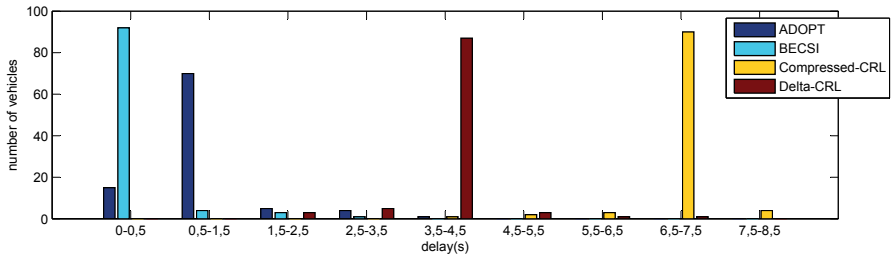

Figure 12: Histogram plot of time delay of the vehicles that receive the CSI depending on the revocation mechanism.

Finally, we also evaluate the overhead introduced by BECSI. BECSI intro-

duces overhead due to the transmission of the value of the hash chain in the control channel. To evaluate this in the CCH channel, we configure the RSUs to transmit this message every second. As expected, the vehicle is receiving messages from the RSU in range every 100 ms; and every second it receives the message M that involves an increase of the incoming throughput of 72 bytes. In this sense, the overhead introduced by the BECSI mechanism is 4% of the toal capacity of the CCH channel.

5 Conclusions

The revocation service is critical to permit efficient authentication in VANETs. Decentralized approaches based on reputation and voting schemes provide mechanisms for revocation management inside the VANET. However, the local validity of the CSI and the lack of support for extending its validity to the global VANET restrain their utilization in the real scenarios. The IEEE 1609.2 standard suggest the use of CRLs to manage the revocation data. However, the tradicional way of issuing CRLs do not fit well in a VANET where huge number of nodes are involved and where several pseudonym certificates are assigned in addition to vehicle identity certificates.

In this paper, we have presented BECSI, a bandwidth efficient certificate status checking mechanism based on the use of a hybrid delta-CRL scheme and MHTs. BECSI introduces an extension to both base-CRL and delta-CRL allowing any non-TTP to act as repository. The main advantage of this *extended-CRL* and *extended-delta-CRL* is that the road-side units and vehicles can build an efficient structure based on an authenticated hash tree to respond to status checking requests inside the VANET, saving time and bandwidth. Thus, vehicles do not have to download the whole CRL but query for the status of the certificate they need to operate with. Moreover, as *extended-delta-CRLs* have a fixed size, BECSI avoids the traditional problem of optimizing the validity windows of delta-CRLs. Thus, the risk of operating with unknown revoked certificates remains constant during the validity interval of the base-CRL, and CAs have the ability to manage this risk by setting the size of the delta-CRLs.

Analytical and simulation results show that allocating a small bandwidth is enough to ensure that vehicles receive CSI responses within few seconds. The performance improvement is obtained at expenses of adding the signed hash tree extension to the standard-CRL. BECSI evaluation shows that not only improves in terms of bandwidth but also in terms of scalability (increase in the number of available repositories) and vulnerability (controlled WOV). In this way, BECSI becomes an offline certificate status validation mechanism as it does not need trusted responders to operate. Therefore, BECSI significantly achieves great efficiency and scalability, especially when deployed in heterogeneous vehicular networks.

Acknowledgments

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

References

[1] R. Bera, J. Bera, S. Sil, S. Dogra, N.B. Sinha, and D. Mondal. Dedicated short range communications (DSRC) for intelligent transport system. In *IFIP International Conference on Wireless and Optical Communications Networks*, pages 1–5, 2006.

[2] *IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments*, May 2008.

[3] IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages. *IEEE Std 1609.2-2006*, pages 1–105, 2006.

[4] M.E. Nowatkowski, J.E. Wolfgang, C. McManus, and H.L. Owen. The effects of limited lifetime pseudonyms on certificate revocation list size in vanets. In *IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the*, pages 380 –383, march 2010.

[5] J.J. Haas, Yih-Chun Hu, and K.P. Laberteaux. Efficient certificate revocation list organization and distribution. *Selected Areas in Communications, IEEE Journal on*, 29(3):595 –604, march 2011.

[6] A. Wasef and Xuemin Shen. Maac: Message authentication acceleration protocol for vehicular ad hoc networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1 –6, 30 2009-dec. 4 2009.

[7] S. Santesson and P. Hallam-Baker. Online Certificate Status Protocol Algorithm Agility. RFC 6277 (Proposed Standard), June 2011.

[8] Maxim Raya, Daniel Jungels, Panos Papadimitratos, Imad Aad, and Jean-Pierre Hubaux. Certificate revocation in vehicular networks. Technical Report LCA-REPORT-2006-006, EPFL, 2006.

[9] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Zhendong Ma, F. Kargl, A. Kung, and J.-P. Hubaux. Secure vehicular communication systems: design and architecture. *Communications Magazine, IEEE*, 46(11):100 –109, November 2008.

[10] ITU-T X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, 2005.

[11] R.C. Merkle. A certified digital signature. In *Advances in Cryptology (CRYPTO89). Lecture Notes in Computer Science*, number 435, pages 234–246. Springer-Verlag, 1989.

[12] Leslie Lamport. Password authentication with insecure communication. *Commun. ACM*, 24:770–772, November 1981.

[13] Maxim Raya and Jean-Pierre 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, pages 11–21, 2005.

[14] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya. Architecture for secure and private vehicular communications. In *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, pages 1 –6, June 2007.

[15] Panagiotis Papadimitratos, Ghita Mezzour, and Jean-Pierre Hubaux. Certificate revocation list distribution in vehicular communication systems. In *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, VANET '08, pages 86–87, 2008.

[16] A. Wasef, Yixin Jiang, and Xuemin Shen. DCS: An Efficient Distributed-Certificate-Service Scheme for Vehicular Networks. *Vehicular Technology, IEEE Transactions on*, 59(2):533 –549, feb. 2010.

[17] Kenneth P. Laberteaux, Jason J. Haas, and Yih-Chun Hu. Security certificate revocation list distribution for vanet. In *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, VANET '08, pages 88–89, 2008.

[18] Chun-I Fan, Ruei-Hau Hsu, and Chun-Hao Tseng. Pairing-based message authentication scheme with privacy protection in vehicular ad hoc networks. In *Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, Mobility '08, pages 82:1–82:7, 2008.

[19] Frederik Armknecht, Andreas Festag, Dirk Westhoff, and Ke Zeng. Cross-layer privacy enhancement and non-repudiation in vehicular communication. In *4th Workshop on Mobile Ad-Hoc Networks (WMAN'07)*, 2007.

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

[21] Tyler Moore, Jolyon Clulow, Shishir Nagaraja, and Ross Anderson. New strategies for revocation in ad-hoc networks. In *Proceedings of the 4th European conference on Security and privacy in ad-hoc and sensor networks*, ESAS'07, pages 232–246, 2007.

[22] A. Wasef and Xuemin Shen. EDR: Efficient Decentralized Revocation Protocol for Vehicular Ad Hoc Networks. *Vehicular Technology, IEEE Transactions on*, 58(9):5214 –5224, nov. 2009.

[23] Maxim Raya, Mohammad Hossein Manshaei, Márk Félegyhazi, and Jean-Pierre Hubaux. Revocation games in ephemeral networks. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 199–210, 2008.

[24] Igor Bilogrevic, Mohammadhossein Manshaei, Maxime Raya, and Jean-Pierre Hubaux. Optimal Revocations in Ephemeral Networks: A Game-Theoretic Framework. In *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt 2010)*, pages 184–193. IEEE, 2010.

[25] Albert Wasef and Xuemin (Sherman) Shen. Emap: Expedite message authentication protocol for vehicular ad hoc networks. *IEEE Transactions on Mobile Computing*, 99(PrePrints), 2011.

[26] Carlos Gañán, Jose L. Muñoz, Oscar Esparza, Jorge Mata-Díaz, Juan Hernández-Serrano, and Juanjo Alins. Coach: Collaborative certificate status checking mechanism for vanets. *Journal of Network and Computer Applications*, (0):–, 2012. (in press).

[27] Carlos Gañán, Jose L. Muñoz, Oscar Esparza, Jorge Mata-Díaz, Juan Hernández-Serrano, and Juanjo Alins. Toward revocation data handling efficiency in vanets. In Alexey Vinel, Rashid Mehmood, Marion Berbineau, Cristina Garcia, Chung-Ming Huang, and Naveen Chilamkurti, editors, *Communication Technologies for Vehicles*, volume 7266 of *Lecture Notes in Computer Science*, pages 80–90. Springer Berlin / Heidelberg, 2012.

[28] Jordi Forné, Jose L. Muñoz, Oscar Esparza, and Francisca Hinarejos. Certificate status validation in mobile ad hoc networks. *Wireless Commun.*, 16:55–62, February 2009.

[29] S. Y. Wang and C. L. Chou. Nctuns tool for wireless vehicular communication network researches. *Simulation Practice and Theory*, 17:1211–1226, 2009.

[30] D.A. Cooper. A model of certificate revocation. In *Fifteenth Annual Computer Security Applications Conference*, pages 256–264, 1999.

[31] Jose L. Muñoz, Oscar Esparza, Carlos Gañán, and Javier Parra-Arnau. Pkix certificate status in hybrid manets. In *WISTP*, volume 5746 of *Lecture Notes in Computer Science*, pages 153–166. Springer, 2009.

[32] Daryl Walleck, Yingjiu Li, and Shouhuai Xu. Empirical analysis of certificate revocation lists. In *Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security*, pages 159–174, 2008.

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

[34] Jason J. Haas, Yih-Chun Hu, and Kenneth P. Laberteaux. Design and analysis of a lightweight certificate revocation mechanism for vanet. In *Proceedings of the sixth ACM international workshop on VehiculAr Inter-NETworking*, VANET '09, pages 89–98, New York, NY, USA, 2009. ACM.

[35] Vikas Taliwal, Daniel Jiang, Heiko Mangold, Chi Chen, and Raja 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, pages 88–88, New York, NY, USA, 2004. ACM.

[36] Chenxi Zhang, Rongxing Lu, Xiaodong Lin, Pin-Han Ho, and Xuemin Shen. An efficient identity-based batch verification scheme for vehicular sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 246 –250, april 2008.