Toward Revocation Data Handling Efficiency in VANETs

Carlos Gañán, Jose L. Muñoz, Oscar Esparza
Jorge Mata-Díaz and Juanjo Alins

Universitat Politècnica de Catalunya (Departament Enginyeria Telemàtica)[**]
{carlos.ganan, jose.munoz, oesparza, jmata,juanjo}@entel.upc.es

Abstract. Vehicular Ad Hoc Networks (VANETs) require some mechanism to authenticate messages, identify valid vehicles, and remove misbehaving ones. A Public Key Infrastructure (PKI) can provide this functionality using digital certificates, but needs an efficient mechanism to revoked misbehaving/compromised vehicles. The IEEE 1609.2 standard states that VANETs will rely on the use of certificate revocation lists (CRLs) to achieve revocation. However, despite their simplicity, CRLs present two major disadvantages that are highlighted in a vehicular network: CRL size and CRL request implosion. In this paper, we point out the problems when using CRLs in this type of networks. To palliate these issues, we propose the use of Authenticated Data Structures (ADS) that allow distributing efficiently revocation data. By using ADS, network entities can check the status of a certificate decreasing the peak bandwidth load in the distribution points.

Keywords: Certification, PKI, Authenticated Data Structures.

1 Introduction

In the last decade, wireless communication between vehicles have drawn extensive attention for their promise to contribute to a safer, more efficient, and more comfortable driving experience in the foreseeable future. This type of communications have stimulated the emergence of Vehicular ad hoc networks (VANETs) which consist of mobile nodes capable of communicating with each other (i.e. Vehicle to Vehicle Communication -V2V communication) and with the static infrastructure (i.e. Vehicle to Infrastructure Communication -V2I communication). To make these communications feasible, vehicles are equipped with on-board units (OBUs) and fixed communication units (road-side units, RSUs) are placed

[**] 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.

along the road. Applying short range wireless technology based on IEEE 802.11, multi-hop communication facilitates information exchange among network nodes that are not in direct communication range [1].

However, the open-medium nature of these networks and the high-speed mobility of a large number of vehicles make necessary the integration of primary security requirements such as authentication, message integrity, non-repudiation, and privacy [2]. Without security, all users would be potentially vulnerable to the misbehavior of the services provided by the VANET. Hence, it is necessary to evict compromised, defective, and illegitimate nodes. The basic solution envisioned to achieve these requirements is to use digital certificates linked to a user by a trusted third party. These certificates can then be used to sign information. Most of the existing solutions manage these certificates by means of a central Certification Authority (CA) [3]. According to IEEE 1609.2 standard [4], vehicular networks will rely on the public key infrastructure (PKI). In PKI, a CA issues an authentic digital certificate for each node in the network. Therefore, an efficient certificate management is crucial for the robust and reliable operation of any PKI. A critical part of any certificate-management scheme is the revocation of certificates.

Regarding the revocation of these certificates, some proposals allow revocation without the intervention of the infrastructure at the expense of trusting other vehicles criteria; and other proposals are based on the existence of a central entity, such as the CA, which is in charge of taking the revocation decision for a certain vehicle. Again, according to the IEEE 1609.2 standard [4], vehicular networks will rely on the existence of a CA. In this sense, it is stated that these networks will depend on certificate revocation lists (CRLs) and short-lived certificates to achieve revocation. CRLs can be seen as black lists that enumerate revoked certificates along with the date of revocation and, optionally, the reasons for revocation.

As the network scale of VANETs is expected to be very large and to protect the privacy of users each vehicle has many temporary certificates (or called pseudonyms), the CRLs are expected to be quite large. Moreover, CRLs have also associated a problem of request implosion, i.e., vehicles may become synchronized around CRL publication instant, as they may request CRL at or near the moment of publication. This burst of requests may cause network congestion that may introduce longer latency in the process of validating a certificate. To reduce the potential network and computational overhead imposed by any CRL distribution mechanism, some optimizations for organizing, storing, and exchanging CRL information have been proposed. In [2, 5], it is proposed a way of compress CRLs using Bloom filters. Their method reduces the size of a CRL by using about half the number of bytes to specify the certificate serial number for revocation. However, the use of this probabilistic structure has associated a false positive rate that diminishes the efficiency of the revocation service.

In this paper, we explore the benefits of using authenticated data structures (ADS), such as binary trees or skip lists, to manage revocation data in VANETS. These structures are a model of computation where untrusted responders answer certificate status queries on behalf of the CA and provide a proof of the validity of the answer to the user. Although VANETs can greatly benefit from the use of ADSs, to the best of our knowledge there has been no proposal of deploying the revocation service by means of an ADS. By using these structures, both CRL issues are palliated: the CA is no longer a bottleneck as there are several responders that act on its behalf; and the revocation data can be checked without downloading the whole CRL.

2 CRLs' problematic in VANETs

As stated in the trial-use standard [4], for a certificate authority (CA) to invalidate a vehicle's certificates, the CA includes the certificate serial number in the CRL. The CA then distributes the CRL so that vehicles can identify and distrust the newly revoked vehicle. The distribution should spread quickly to every vehicle in the system.

However, the distribution itself poses a great challenge due to the size of the CRL. As a CRL is a list containing the serial numbers of all certificates issued by a given certification authority (CA) that have been revoked and have not yet expired, its distribution causes network overhead. Moreover, the CRL size increases dramatically if only a small portion of the OBUs in the VANET is revoked. To have an idea of how big the CRL size can be, consider the case where 1% of the total number of the OBUs in the United States is revoked. 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 [5] 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. In the United States alone, 255,917,664 "highway" registered vehicles were counted in 2008, of which 137,079,843 passenger cars [6]. In this case, the CRL would contain around 100 billion revoked certificates. Assuming that certificates can be identified by a 16 byte fingerprint (the size of one AES block), the CRL size would be of 1,7 TB approximately. Only the amount of memory necessary to storage this CRL makes it impossible its deployment. Therefore, the CRL size has to be reduced.

The CRL size can be reduced by using regional CAs. However, there appears a trade-off between the size of the CA region and size of the CRL, as well as the management complexity of the entire PKI system for VANETs. The least complicated region to manage would be a single large area, such as the entire United States, with a single CA responsible for every certificate and pseudonym. However, this gives place to CRLs of several terabytes. Therefore, it is necessary to divide the CRL information according to regional areas. In this sense, if we divide the entire United States by cities (i.e. 10,016 cities according to the U.S. census bureau), the CRL size is reduced to around 170 Mbytes. Using the 802.11a protocol to communicate with RSUs in range, vehicles could have between 10-30 Mbps depending on the vehicle's speed and the road congestion. Therefore, in the best case a vehicle will need more than 45 seconds to download the whole CRL. Under non-congested conditions, any vehicle should be able to contact the infrastructure for more than 45 seconds, and therefore download the CRL. In scenarios where vehicles are not able to keep a permanent link with the infrastructure for this amount of time, techniques such as Bloom filter or Digital Fountain Codes could be used to download the CRL. Therefore, though the problem of having a huge CRL is mitigated by the use of such techniques, the restraints imposed by the distribution affect the freshness of the revocation data.

A direct consequence of this significant time to download a CRL is that a new CRL cannot be issued very often, so its validity period has to be shortened. This validity period directly determines how often a vehicle has to update the revocation information. Therefore, the validity period of the CRL is critical to the bandwidth consumption. In this context, it appears another trade-off between the freshness of revocation information and the bandwidth consumed by downloading CRLs. Large validity periods will decrease the network overhead at expenses of having outdated revocation information. Small validity periods will increase the network overhead but users will have fresh information about revoked certificates. As CRLs cannot be issued every time there is a new revoked certificate, vehicles will be operating with revocation information that is not comprehensive. Therefore, they will be taking certain risk of trusting a certificate that could be potentially revoked.

3 Using Authenticated Data Structures for certificate revocation in VANETs

By replicating revocation data at untrusted responders near users, VANETs can enhance its performance but that replication causes a major security challenge. Namely, how can a vehicle verify that the revocation data replicated at the RSUs are the same as the original from the CA? A simple mechanism to achieve the authentication of replicated revocation data consists of having the digitally sign each revocation entry and replicating the CA signature too. However, in VANETs where the revocation data evolves rapidly over time, this solution is inefficient. To achieve higher communication and computation efficiency, we propose the use of authenticated data structures (ADS) to handle the revocation service in VANETs. ADSs are a model of computation where untrusted responders answer certificate status queries on behalf of the CA and provide a proof of the validity of the answer to the user. In this section, first we introduce the architecture necessary to adopt ADSs. Then, we describe different ADSs and their main benefits.

3.1 System Architecture

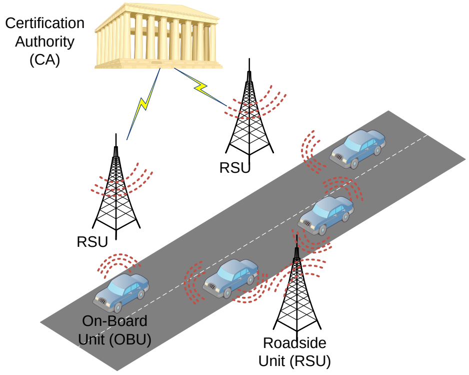


Fig. 1. System Architecture.

The system architecture to support ADSs consists in an adaptation of a PKI system to the vehicular environment. The ADS model involves a structured collection $\mathcal{R}$ of revoked certificates and three parties: the certification authority (CA), the road side units (RSUs), and the vehicles. A repertory of query operations and optional update operations are assumed to be defined over $\mathcal{R}$. These three parties present a hierarchical architecture (see Fig. 1) which consists of three levels: the CA is located at level 1, as it is the top of the system. RSUs are located at level 2. Finally, the on-board units (OBUs) are located at the bottom of the hierarchy. Note that without loss of generality we consider a single the central trusted authority at the root, but it could be further divided into different state level trusted authorities and additionally a group of city level trusted authorities can be placed under every state authority.

The main tasks of each entity are:

1. The CA is responsible for generating the set of certificates that are stored in each OBU. It is also responsible for holding the original version of $\mathcal{R}$ 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 the CA is the only trusted entity within the network. Whenever an update is performed on $\mathcal{R}$, the CA produces structure authentication information, which consists of a signed time-stamped statement about the current version of $\mathcal{R}$.
2. 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. RSUs maintain a copy of $\mathcal{R}$. They interact with the CA by receiving from the CA the updates performed on $\mathcal{R}$ together with the associated structure authentication information. RSUs also interact with vehicles by answering queries on $\mathcal{R}$ posed by the vehicles. In addition to the answer to a query, RSUs also return answer authentication information, which consists of (i) the freshest structure authentication information issued by the CA; and (ii) a proof of the authenticity of the answer. If the CA considers that an RSU has been compromised, the CA can revoke it.
3. OBUs are in charge of storing all the certificates that a vehicle possesses. An OBU has abundant resources in computation and storage and allows any vehicle to communicate with the infrastructure and with any other vehicle in its neighborhood. OBUs pose queries on $\mathcal{R}$, but instead of contacting the CA directly, it contacts the RSU in range. However, OBUs only trust the CA and not the RSU about $\mathcal{R}$. Hence, it verifies the answer from the RSU using the associated answer authentication information.

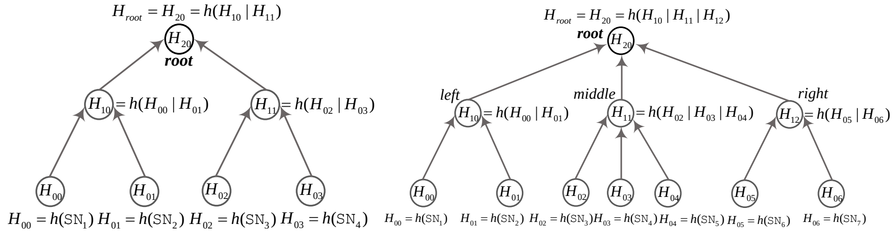
3.2 System Requirements

- *Low computational cost*: The computations performed internally by each entity (CA, RSU, and OBU) should be simple and fast.
- *Low communication overhead*: CA-to-RSU communication (update authentication information) and RSU-to-OBU communication (answer authentication information) should be as small as possible.
- *High security*: the authenticity of the answers given by a RSU should be verifiable.

3.3 Authenticated Data Structures

Several ADSs have been proposed in the literature (mainly in the context of data base management) that fulfill the aforementioned requirements. In this section, we describe a repertoire of ADSs and to what extent they are capable of improving the revocation service.

Merkle Hash trees A Merkle hash tree (MHT) [7] is essentially a tree structure that is built with a collision-resistant hash function to produce a short cryptographic description of $\mathcal{R}$. The leaf nodes hold the hash values of the data of interest, i.e., the serial number of the revoked certificates (SN_1,SN_2,. . .,SN_n); and the internal nodes hold the hash values that result from applying the hash function 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.


$H_{root} = H_{20} = h(H_{10} | H_{11})$
H_{20}
root
H_{10} $= h(H_{00} | H_{01})$
H_{11} $= h(H_{02} | H_{03})$
H_{00}
H_{01}
H_{02}
H_{03}
$H_{00} = h(\mathrm{SN}_1)$ $H_{01} = h(\mathrm{SN}_2)$ $H_{02} = h(\mathrm{SN}_3)$ $H_{03} = h(\mathrm{SN}_4)$
$H_{root} = H_{20} = h(H_{10} | H_{11} | H_{12})$
root
H_{20}
left
middle
right
H_{10} $= h(H_{00} | H_{01})$
H_{11} $= h(H_{02} | H_{03} | H_{04})$
H_{12} $= h(H_{05} | H_{06})$
H_{00}
H_{01}
H_{02}
H_{03}
H_{04}
H_{05}
H_{06}
$H_{00} = h(\mathrm{SN}_1)$ $H_{01} = h(\mathrm{SN}_2)$ $H_{02} = h(\mathrm{SN}_3)$$H_{03} = h(\mathrm{SN}_4)$ $H_{04} = h(\mathrm{SN}_5)$ $H_{05} = h(\mathrm{SN}_6)$ $H_{06} = h(\mathrm{SN}_7)$

(a) (b)

Fig. 2. Sample trees (a) MHT (b) 2-3.

A sample MHT is presented in Figure 2(a). The authentication of an element is performed using a verification path, which consists of the sibling nodes of the nodes on the path from the leaf associated with the element to the root of the tree. The root value is signed and the collision-resistant property of the hash function is used to propagate authentication from the root to the leaves. This construction is simple and efficient and achieves signature amortization, where only one digital signature is used for signing a large collection of data. The hash tree uses linear space and has $O(log\ n)$ (where n denotes the number of revoked certificates) proof size, query time and verification time. An ADS based on hash trees can also achieve $O(log\ n)$ update time.

2-3 trees A standard 2-3 tree [8] is a tree where all leaves are at the same height and each node (except leaves) has two or three children. It has the nice property that leaf removal and insertion incur only logarithmic complexity because these operations only involve the nodes related to the path from the relevant leaf to the root.

Each leaf of such a 2-3 tree stores an element of set $\mathcal{R}$, and each internal node stores a one-way hash of its children's values. Thus, the CA-to-RSU communication is reduced to $O(1)$ entries, since the CA sends insert and remove instructions to the RSUs, together with a signed message consisting of a timestamp and the hash value of the root of the tree. RSUs respond to a membership query for an element SN_i as follows: if SN_i is in $\mathcal{R}$, then RSUs provide the path from the leaf storing SN_i to the root, together with all the siblings of the nodes on this path; else (SN_i is not in$\mathcal{R}$), RSUs provide the leaf-to-root paths from two consecutive leaves storing SN_j and SN_k such that $j < i < k$, together with all siblings of the nodes on these paths. By tracing these paths, OBUs can recompute the hash values of their nodes, ultimately recomputing the hash value for the root, which is then compared against the signed hash value of the root for authentication. As with MHTs, these trees achieve $O(log\ n)$ proof size, query time, update time and verification time.

One-way accumulator One-way accumulator (OWA) functions [9] allow a CA to digitally sign a collection of objects as opposed to a single one. The main advantage of this approach is that the validation of a response takes constant time and requires computations simple enough to be performed in resource-constrained devices. This type of ADS achieves a tradeoff between the cost of updates at the CA and queries at the RSUs, with updates taking $O(k + log(\frac{n}{k}))$ time and queries taking $O(\frac{n}{k})$ time, for any fixed integer parameter $1 \leq k \leq n$. For instance, one can achieve $O(\sqrt{n})$ time for both updates and queries.

Skip Lists Skip lists [10] are probabilistic ADSs that provide an alternative to balanced tree. Skip lists are sorted linked lists with extra links, designed to allow fast search in $\mathcal{R}$ by taking "shortcuts". The main idea is to enhance linked lists, which connect each element in the data sequence to its successor, by also connecting some elements to successors further down the sequence. Roughly half of the elements have links to their two-hop successor, roughly a quarter of the elements have links to their four-hop successor, and so on. As a result, during traversal from SN_i to element SN_j, the traversal path follows repeatedly the longest available link from the current element that does not overshoot the destination SN_j, and thereby reaches SN_j in fewer steps than would be possible by just traversing every intervening element between SN_i and SN_j. Compared with balanced trees, a skip list presents the following benefits:

- It is easy to implement and practically efficient in search, especially update time.
- It is space compact, where space is allocated when needed, while empty space is preserved in balanced tree.
- It is main memory index, while balanced tree are disk-based index.

Finally, Table 1 shows a comparison of the asymptotic performance of the main ADS versus traditional revocation mechanisms such as CRL or OCSP. Note that with ADSs, the revocation service can be greatly improved both in computation and communication overhead.

method	**space**	**update time**	**update size**	**query time**	**query size**	**verifying time**
CRL	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
OCSP	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(1)$
MHT	$O(n)$	$O(log\ n)$	$O(1)$	$O(log\ n)$	$O(log\ n)$	$O(log\ n)$
2-3 tree	$O(n)$	$O(log\ n)$	$O(1)$	$O(log\ n)$	$O(log\ n)$	$O(log\ n)$
Skip Lists	$O(n)$	$O(log\ n)$	$O(1)$	$O(log\ n)$	$O(log\ n)$	$O(log\ n)$
OWA	$O(n)$	$O(k + log(\frac{n}{k}))$	$O(k)$	$O(\frac{n}{k})$	$O(1)$	$O(1)$

Table 1. Comparison of the main ADS vs traditional revocation mechanisms.

3.4 Certificate Status Validation Protocol

The certificate status validation protocol consists in three stages.

1. *Revocation Service Setup*: The CA creates a CRL by appending the serial number of any revoked certificate. Then, it computes the corresponding ADS from the set $\mathcal{R}$ of revoked certificates contained in the

CRL. Once the ADS is computed, the CA signs the resulting timestamped digest of the data structure, i.e., a collision resistant succinct representation of the data structure. The digest is transmitted to all the RSUs via a secure wireline together with the corresponding CRL. RSUs can either implement a push or pull protocol to transmit the digest to the vehicles in range.

2. *Certificate Status Updating*: Depending on the CA's policy, when an update is necessary, the CA recomputes the ADS and generates a new signed digest that is transmitted to the RSUs. Note that depending on the ADS, the data structure should be computed again or only update and delete operations should be performed. The new ADS is transmitted to the RSUs again, so that they could answer to validation queries.
3. *Certificate Status Querying*: OBUs query any RSU in range about the status of a particular certificate (SN_i). If $SN_i \in \mathcal{R}$, then the RSU computes the path necessary to allow OBUs to compute the digest and check that it matches the signed digest. If $SN_i \notin \mathcal{R}$, then the RSU computes the path of two consecutive certificates in $\mathcal{R}$ and transmit them to the requesting OBU. This OBU can then recompute the digest for both revoked certificates and be sure that $SN_i \notin \mathcal{R}$.

4 Evaluation

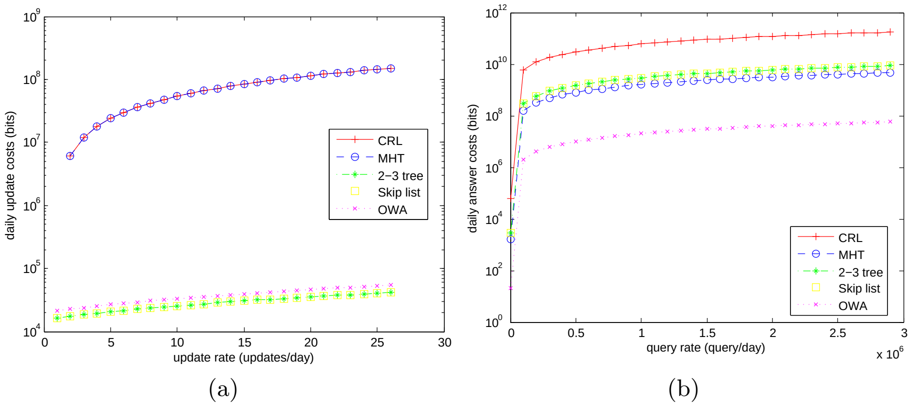
In the following, we compare the communication costs of using ADSs with the tradition CRL mechanism. To that end we define a set of parameters (see Table 2).

Parameter	**Meaning of the parameter**
N	Total number of certificates ($n = 3,000,000$)
k	Average number of certificates handled by a CA ($k = 30,000$)
p	Percentage of revoked certificates($p = 0.1$)
q	Number of certificate status queries issued per day ($q = 3,000,000$)
T	Number of updates per day ($T = 1$)
s_{SN}	Size of a serial number ($s_{SN} = 20$)
s_{sig}	Size of a signature ($s_{sig} = 1,000$)
s_{hash}	Size of the hash function ($s_{hash} = 128$).

Table 2. Notation

Using this notation, the CRL daily update cost is $T \cdot n \cdot p \cdot s_{SN}$ as each CA sends the whole CRL to the corresponding RSUs in each update.

The CRL daily query cost is $q \cdot p \cdot k \cdot s_{SN}$ as for every query the RSU sends the whole CRL to the querying OBU. When using ADS, these costs are drastically reduced. Note that no matter the type of ADS, OBUs do not have to download the whole CRL, and they only download status information about the certificate they want to operate with. Regarding MHTs, the RSUs have to recompute the tree in each update, so that daily update cost is $T \cdot n \cdot p \cdot s_{SN}$. However, to answer an OBU's query the RSU only needs to send up to $1 + log_2(pk)$ numbers, resulting in $q \cdot s_{hash}(1+log_2(pk))$ bits. In the case of 2-3 trees, to update the directory, the CA sends difference lists of total daily length of $\frac{n \cdot p \cdot s_{SN}}{365} + T \cdot s_{sig}$; and answer to OBUs' queries results in $2 \cdot q \cdot s_{hash} \cdot log_2(pk)$ bits. Similarly, skip lists need $2log_2\lceil pk \rceil$ number to answer an OBU's query and the same update cost than the 2-3 tree. With OWAs, the size of answer are drastically reduced to roughly s_{sig}, and the update cost depends on the accumulator configuration. We use Matlab R2011b to evaluate these costs.



(a) (b)

Fig. 3. (a) Daily CA-to-RSU update costs vs. update rate, (b) RSU-to-OBU query cost vs query rate.

Note that the costs will vary mainly depending on the total number of revoked certificates, the update rate and the number of queries. Figure 3(a) shows how the CA-to-RSU update communication costs of the different revocation mechanisms depend on the update rate (all other parameters are held constant). Note that any ADS is much more robust and efficient than CRL, even allowing once per hour updates. Regarding the query costs, as ADSs have smaller proof to validate the status of a certificate they provide a more bandwidth efficient solution than CRL (see Fig. 3(b)).

5 Conclusions

In this paper, we consider the problem of certificate authentication and revocation in VANETs. We have proposed the use of authenticated data structures to handle the revocation service over VANETs. After discussing the issues of deploying CRLs in these environments, we show that ADSs are more robust to changes in parameters, and allow higher update/query rates than traditional revocation mechanisms. In addition, the adoption ADS reduces both the communication and the computational overhead in the OBUs. For our future work, we will investigate the use of mobile repositories under the context of the proposed schemes.

References

1. D. Jiang and L. Delgrossi. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040, May 2008.
2. 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.
3. P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya. Architecture for secure and private vehicular communications. In *Telecommunications, 2007. 7th International Conference on ITS*, pages 1 –6, June 2007.
4. 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.
5. 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 InterNETworking*, VANET '09, pages 89–98, New York, NY, USA, 2009. ACM.
6. Bureau of Transportation Statistics U.S. Department of Transportation. Number of u.s. aircraft, vehicles, vessels, and other conveyances. `http://www.bts.gov/publications/national_transportation_statistics/html/table_01_11.html`, 2009. [Online; accessed 31-July-2011].
7. 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.
8. M. Naor and K. Nissim. Certificate Revocation and Certificate Update. *IEEE Journal on Selected Areas in Communications*, 18(4):561–560, 2000.
9. Josh Benaloh and Michael de Mare. One-way accumulators: a decentralized alternative to digital signatures. In *Workshop on the theory of cryptographic techniques on Advances in cryptology*, EUROCRYPT '93, pages 274–285, 1994.
10. William Pugh. Skip lists: a probabilistic alternative to balanced trees. *Commun. ACM*, 33:668–676, June 1990.