

# Eviction of Misbehaving and Faulty Nodes in Vehicular Networks

M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux

**Abstract**—Vehicular Networks (VNs) are emerging, among civilian applications, as a convincing instantiation of the mobile networking technology. However, security is a critical factor and a significant challenge to be met. Misbehaving or faulty network nodes have to be detected and prevented from disrupting network operation, a problem particularly hard to address in the life-critical VN environment. Existing networks rely mainly on node certificate revocation for attacker eviction, but the lack of an omnipresent infrastructure in VNs may unacceptably delay the retrieval of the most recent and relevant revocation information; this will especially be the case in the early deployment stages of such a highly volatile and large-scale system. In this paper, we address this specific problem. We propose protocols, as components of a framework, for the identification and local containment of misbehaving or faulty nodes, and then for their eviction from the system. We tailor our design to the VN characteristics and analyze our system. Our results show that the distributed approach to contain nodes and contribute to their eviction is efficiently feasible and achieves a sufficient level of robustness.

**Index Terms**—vehicular networks, misbehavior detection, certificate revocation

## I. INTRODUCTION

RECENT research initiatives supported by governments and car manufacturers seek to enhance the safety and efficiency of transportation systems. Vehicular networks lie at the core of these efforts. Vehicular network nodes, that is, vehicles and Road-Side Infrastructure Units (RSUs) will be equipped with sensing, processing, and wireless communication modules. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication will enable safety applications that provide warnings about accidents, traffic conditions (e.g., congestion, emergency braking) and other events.

Integrating security mechanisms into the VNs is critical for their deployment: their rich functionality and services can be otherwise abused, jeopardizing the safety of vehicles, drivers, and passengers, as well as the efficiency of the transportation system. A number of research contributions analyze vulnerabilities [1], [2], outline architectural components, requirements, and design principles [3], and propose specific mechanisms [4], [5], [6], [7], [8].

Manuscript received February 22, 2007; revised June 12, 2007. Part of this work was funded by the EU project SEVECOM (<http://www.sevecom.org>) and done during I. Aad's and D. Jungels' stay at EPFL.

M. Raya, P. Papadimitratos, and J.-P. Hubaux are with LCA Lab, EPFL, Switzerland (e-mail: [firstname.lastname@epfl.ch](mailto:firstname.lastname@epfl.ch)).

I. Aad is with Future Networking Lab, DoCoMo Euro-Labs, Germany (e-mail: [aad@docomolab-euro.com](mailto:aad@docomolab-euro.com)).

D. Jungels is with HITEC Luxembourg S.A. (e-mail: [daniel.jungels@a3.epfl.ch](mailto:daniel.jungels@a3.epfl.ch)).

Digital Object Identifier 10.1109/JSAC.2007.0710xx.

The presence of an authority, which we denote as the Certification Authority (CA), is implied or mandated in practically all the research efforts concerned with securing VNs. Rigid identity and credential management processes for vehicles and drivers have long been in place; accountability and attribution of liability will continue to be crucial; and access control mechanisms will be necessary. Without the appropriate certificates and cryptographic keys, nodes are essentially unable to participate in the network operation. Nevertheless, the possession of a certificate does not guarantee that its holder will provide correct information: a node can simply inject faulty data (e.g., alerts, warnings, coordinates) while complying with the implemented protocols. Safeguarding the system against such faulty or compromised nodes is crucial for its robustness. Hence the need for the eviction of misbehaving nodes. A typical approach for achieving this is the revocation of node certificates; once this is done, messages from these nodes will be ignored.

Timely access to revocation information is a particularly hard problem in VNs. The road-side infrastructure can act as the gateway of the CA to the network, distributing the latest Certificate Revocation Lists (CRLs) [9]. The lack of an omnipresent road-side infrastructure, especially in the early deployment stages, and the huge scale of the VNs are obstacles to the application of traditional certificate revocation schemes. Moreover, unless a node is revoked for administrative reasons (e.g., the vehicle owner did not renew its registration), how can the authority obtain and validate sufficient evidence that a node is faulty or compromised? Thus, an additional challenge is how non-misbehaving nodes can be protected until they obtain the revocation information regarding misbehaving nodes.

Our contributions in this paper address these problems. We propose the combination of (i) infrastructure-based revocation protocols, the Revocation of the Trusted Component (RTC) and Revocation using Compressed Certificate Revocation Lists (RC<sup>2</sup>RL), (ii) a Misbehavior Detection System (MDS) enabling the neighbors of a misbehaving or faulty node to detect its deviation from normal behavior, and initiate (iii) a Local Eviction of Attackers by Voting Evaluators (LEAVE) protocol to safeguard the system operation, until the attacker is revoked by the CA, partially or fully based on the evidence LEAVE provides.

We emphasize however that no group of nodes has the power to revoke another node. The CA is the sole entity with the right to initiate a revocation protocol. This design choice ensures resilience to collusion attacks, retains accountability, and yet equips nodes with a rapid reaction and self-protection tool. Indeed, our performance evaluation results show that a

high percentage of the attacker's neighbors can be alerted of its misbehavior, despite the very short contact time between the protocol participants.

The rest of this paper is organized as follows. We first describe the system and adversary models in Sec. II. We provide an overview of our scheme in Sec. III and then present its components in further detail in Sections IV-VI. We evaluate our scheme in Sec. VII. We survey related work in Sec. VIII, before we conclude.

## II. SYSTEM MODEL

Drawing from the analogy with existing administrative processes and automotive authorities (e.g., city or state transit authorities), a large number of CAs will exist. Each of them is responsible for the identity management of all vehicles registered in its *region* (national territory, district, county, etc.). Vehicles are registered with exactly one CA. Each node has a unique identity  $V$  and a pair of *private* and *public* cryptographic keys,  $PrK_V$  and  $PuK_V$ , respectively, and is equipped with a certificate  $Cert_{CA}\{V, PuK_V\}$  issued by the CA. The vehicle may have several keys for privacy reasons [7] but this does not affect the operation of the proposed mechanisms (because revocation and eviction apply actually to the keys of a vehicle) and is out of scope of this paper.

Messages are transmitted periodically, e.g., every 0.3 s for *safety messages*, or triggered by in-vehicle or network events. Most of the traffic is broadcasted to limited regions of the network; these regions are determined by the corresponding applications. All safety-related messages include the time and geographical coordinates (obtained by a positioning service, such as the widely available GPS) of the sender, in addition to other application-specific information. In addition, each message is signed and accompanied by the sender's certificate. The feasibility of asymmetric cryptography, namely digital signatures, in VNs, has been shown in prior work [10].

Safety messages that need to propagate across multiple hops (and perhaps have the originator's signature, coordinates and time intact as they propagate) are signed and include the coordinates and timestamp of the last relaying node. This ensures the freshness of the information and limits the propagation of illegitimate information. A received safety message is discarded if the difference between its timestamp and the timestamp of the receiver is larger than a system-specific constant that accounts for the maximum clock drift and one-hop transmission, propagation and processing delays. Moreover, a message is discarded (at a receiver) if the coordinates of its sender/relay, as reported in the message, indicate that the receiver is outside the sender's maximum nominal wireless communication range (accounting for location information inaccuracies). This validation is applied only on a per-hop manner.

At the data link layer, the Dedicated Short Range Communications (DSRC) protocol [11], currently being standardized as IEEE 802.11p, provides transmission ranges of typically 300 to 1000m, with data rates in the 6-27 Mbps range. In this paper, we assume that 802.11p is used, unless noted otherwise. Beyond DSRC, vehicular networks can leverage on other wireless communication technologies, such as the

(licensed-frequency) existing cellular networks, broadband wireless (e.g., WiMax), or low-speed radio broadcast systems used today for traffic information.

We denote a subset of the network nodes as the infrastructure, comprising the RSUs (i.e., short-range DSRC *base stations*) and *mobile units*. The latter include public safety vehicles (e.g., highway assistance and fire-fighting vehicles), police vehicles, aerial vehicles (e.g., police helicopters), and public transport vehicles (e.g., buses, trams). In our context, these nodes can be used, for example, to disseminate CRLs. Infrastructure nodes serve as the gateway of the CA to/from the VN; the connection of the CA to the static infrastructure nodes is over wireline secure links. We note however that accessibility of the CA from the VN is not assumed to be guaranteed at all times.

Many vehicles are already equipped with hardware and firmware components, such as speed limiters, tachographs, and *event data recorders (EDRs)*, which are considered critical by manufacturers and legislators. We assume that nodes are equipped with a Trusted Component (TC), i.e., tamper-resistant hardware and firmware. The role of the TC is two-fold: (i) it stores all cryptographic material and prevents its exposure to the on-board computer; (ii) it performs all cryptographic operations. This assumption, although seemingly strong, has been previously justified in the context of VNs [7]. In addition, recent advances in TC design [12] suggest that sufficiently performing, yet reasonable priced, devices will be available on the market.

### A. Adversary Model

We term as an *adversary* or *attacker* any node that deviates from the legitimate VN protocols. Nodes can also be *faulty* due to failures of their equipment. A detailed discussion of adversary and fault models is given in [3]. Any of these attacks or faults, or combinations thereof, can affect the VN-enabled applications. We also refer to adversaries as *misbehaving nodes*. As our proposed mechanisms apply to both misbehaving and faulty nodes, we will use both terms interchangeably in the remainder of this paper without losing the generality of the solutions.

In addition, the information-oriented operation of VNs, with their diverse data types, makes *false information dissemination* a very effective attack, compared to deviations from the networking protocols. In fact, it would suffice for an adversary to manipulate the sensory inputs rather than compromise the protocol stack and the computing platform [3]. It is also possible that an attacker controls incoming communication, i.e., selectively erasing messages received by its on-board platform.

We emphasize that we are concerned with misbehaving nodes equipped with valid credentials, because they can effectively abuse the system. An essential assumption we make is *the existence of an honest majority in the attacker's neighborhood* (defined in Sec. VI-B). As we will show later, this allows vehicles to rely on their honest neighbors in order to evict attackers. This assumption (also elaborated in [13], [3]) may appear limiting; but it is reasonable if we consider the existing transportation systems where the actual percentage of attackers is very low.

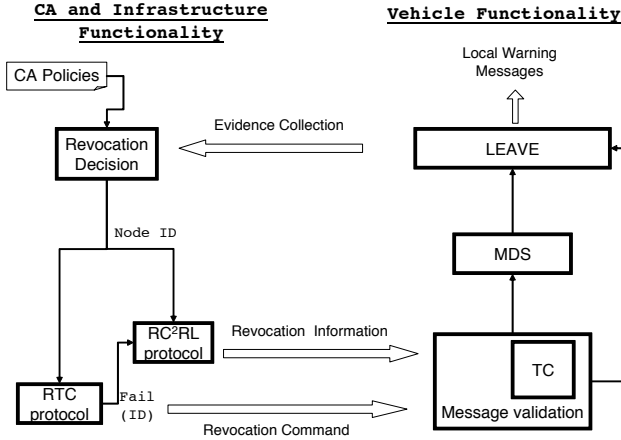


Fig. 1. Detection and Eviction Scheme Overview.

### III. SCHEME OVERVIEW

Our scheme consists of the following basic components: (i) the centralized revocation of a node by the CA, (ii) the local detection of misbehavior, performed individually by each node and (iii) a distributed, localized protocol for the eviction of an attacker by its neighboring nodes. The scheme with its components is illustrated in Fig. 1.

We propose two methods for misbehaving node revocation, initiated by the CA. The first one, **RTC (Revocation of the TC)**, described in Sec. IV-A), leverages on the presence of a TC unit on board the vehicle. The CA determines that a vehicle  $V$  must be revoked and, with the help of the road-side infrastructure, initiates a two-party end-to-end protocol with  $TC_V$ , the trusted component of  $V$ . The CA instructs the TC to erase all cryptographic material (e.g., keys) it stores and halt its operation upon completion of the protocol. Essentially, this protocol “kills” the TC, depriving the misbehaving node from its cryptographic keys, and thus ensuring that all its messages are ignored by all other correct nodes.

However, RTC is not robust against a sophisticated adversary that controls the communication link between the CA and the TC. If the CA fails in executing RTC (detected by the lack of an acknowledgment), it will revert to the distribution of the revocation information, namely, a CRL, to the VN. This way, the CA invalidates credentials before the end of their lifetime. But the size of CRLs will grow with the size of the VN and hence is not scalable. To adapt this approach to the VN scale, we propose the **RC<sup>2</sup>RL (Revocation using Compressed Certificate Revocation Lists)** protocol (Sec. IV-B), with Compressed CRLs (C<sup>2</sup>RLs) being shorter than traditional CRLs by means of Bloom filter compression.

The timely and efficient distribution of revocation information across the VN is the primary means of revoking misbehaving nodes. However, to design a robust and efficient system capable of progressively isolating misbehaving nodes before this information becomes available, we propose the use of a localized **MDS (Misbehavior Detection System)** and the **LEAVE (Local Eviction of Attackers by Voting Evaluators)** protocol.

MDS, discussed further in Sec. V, is an essential enabler of LEAVE. Each node uses its own sensory inputs (including time and location), messages received from its neighbors (assuming an honest majority), and a set of *evaluation rules*, to classify safety messages received from a given node as faulty or correct. Messages that are outdated (aged), received beyond their expected area of propagation, or contradictory to the node’s own state<sup>1</sup> are considered false. Their senders, as long as they are neighbors of the node running MDS, are also tagged as misbehaving. Then, their identity is passed to LEAVE.

The main principle of LEAVE, detailed in Sec. VI, is simple: the neighbors of the misbehaving vehicle *temporarily* “evict” it. In contrast to RTC and RC<sup>2</sup>RL, LEAVE is not a revocation protocol, but rather a *collective warning system against misbehaving nodes*. Upon detecting an attacker, vehicles broadcast *warning* messages to all vehicles in range, so that the sharing of information improves the effectiveness of the stand-alone detection systems. Moreover, such warnings can be invaluable when vehicles receive them before being able to observe the misbehaving node themselves. We clarify that the notion of neighborhood is different for MDS and LEAVE. In the first case, it includes all one-hop neighbors of the vehicle running the MDS. LEAVE, as further detailed in Sec. VI-B, elects a subset of this neighborhood; this subset depends on both the vehicle running the MDS and the attacker.

The eviction of an attacker by its neighbors is temporally limited to the duration of contact between the attacker and its neighbors running LEAVE. But once enough evidence against the attacker is gathered, the CA can initiate one of the previously described revocation protocols. Recall that the CA is the only system entity entitled to revoke keys (due to all the related administrative responsibilities and costs). In this paper, we do not consider the CA decision process for node revocation, as a number of legal and policy aspects are involved. In addition, the reasons for revocation are largely orthogonal to the operation itself and can include administrative procedures (e.g., change of registration domain), cryptographic material compromise (e.g., a private key was detectably disclosed) or, as mentioned above, node misbehavior for which the CA obtains sufficient evidence.

### IV. REVOCATION PROTOCOLS

#### A. Revocation of the Trusted Component (RTC)

When the CA decides to revoke a vehicle  $V$ , it first uses RTC (Fig. 2): The CA generates a revocation message that contains  $V$ ’s identity, encrypted with  $V$ ’s public key  $PuK_V$ , and a timestamp  $T$ ; the message is signed by the CA. Thus,  $TC_V$  and the RSUs that forward the message can verify its authenticity and freshness. The message format is:

$$CA \xrightarrow{RSU} TC_V : E_{PuK_V}(V), T, Sig_{CA}[E_{PuK_V}(V), T]$$

where  $E_{PuK_V}()$  denotes encryption with public key  $PuK_V$ .

There are several options for channeling this message to the  $TC_V$ . The first choice would be to route it to the RSU closest

<sup>1</sup>For example, a traffic jam message received when the node’s velocity in the allegedly jammed area is well above the velocity expected for a traffic jam.

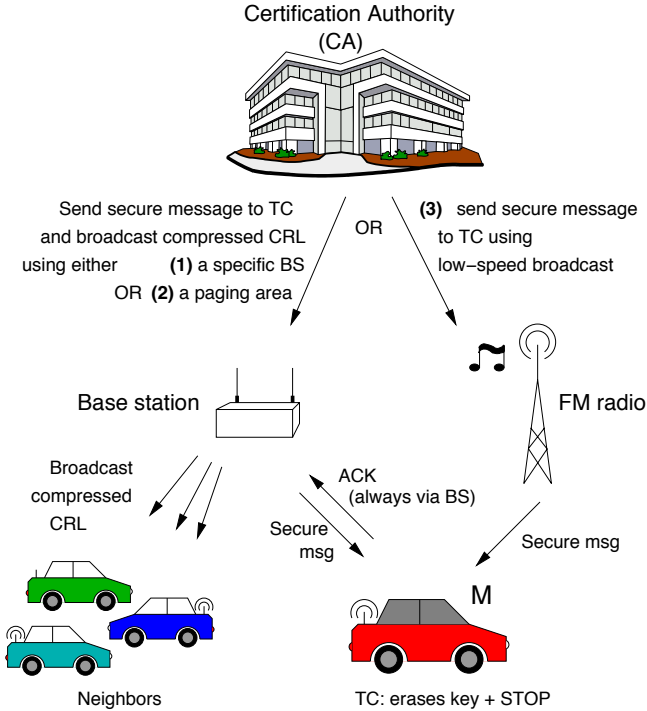


Fig. 2. Revocation of the Trusted Component (RTC) and Revocation using Compressed Certificate Revocation Lists (RC<sup>2</sup>RL).

to the concerned vehicle, if its location is known to the CA. Otherwise the CA defines a paging area consisting of several RSUs in the region of the vehicle's most recent locations (trajectory extrapolation based on the vehicle's expected speed and acceleration can be useful in determining the paging area). If all else fails, the CA can use other distribution media mentioned in Section II, such as low-speed radio broadcast.

When  $TC_V$  receives the RTC message, it immediately erases the cryptographic key and stops signing VN messages. It sends back a timestamped and signed acknowledgment, as soon as it comes within range of a RSU:

$$TC_V \xrightarrow{RSU} CA : ACK, T, Sig_{PrK_V}[ACK, T]$$

If the vehicle  $V$  is an attacker capable of blocking messages destined to its TC, the CA will receive no acknowledgement and thus will detect the failure of RTC. It will then revert to the RC<sup>2</sup>RL protocol discussed in the next subsection.

### B. RC<sup>2</sup>RL (Revocation using Compressed Certificate Revocation Lists)

As CRLs contain very little redundancy, they cannot be efficiently compressed using normal lossless methods. We therefore use Bloom Filters [14], a special form of lossy compression, to generate C<sup>2</sup>RLs (Compressed CRLs) that the CA signs and broadcasts using one of the previously mentioned distribution methods. Bloom filters provide a probabilistic data structure used to test whether an element is a member of a set. They are characterized by a configurable rate of false positives and no false negatives. This ensures that the CA can efficiently revoke all targeted nodes while keeping false revocations within acceptable error margins. A more detailed

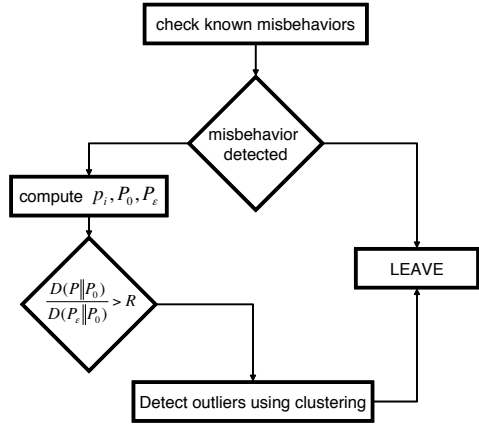


Fig. 3. Misbehavior Detection System (MDS) operation.

explanation of Bloom filters and their application to revocation in VNs can be found in Appendix I.

## V. MISBEHAVIOR DETECTION SYSTEM

As explained in Sec. II-A, it may be more beneficial for the adversary in VNs to tamper with the data transferred by protocols rather than with the protocols themselves. Hence, the *Misbehavior Detection System* (MDS) should rely not only on the protocol-specific actions of nodes but also on the data these nodes provide. Similarly to Intrusion Detection Systems (IDS) [15], we can distinguish between two types of misbehavior in VNs:

- 1) *Known misbehaviors* that can be identified by monitoring specific parameters of node or network behavior. For example, several tests for position verification in georouting protocols are proposed in [16].
- 2) *Data anomalies* that do not follow any known pattern. This is often the case when the adversary modifies or injects safety messages according to its specific needs.

Standard IDS techniques detect data anomalies by monitoring specific metrics, comparing the actual metric values to expected values, and by thresholding the deviation to detect attacks. In VNs, expected values often are not known in advance and do not fit a given model. For example, the traffic congestion varies considerably depending on the road and the time of the day. A highly adaptive approach consists in comparing the behavior of each node to the average behavior of the other nodes (including the node running the MDS), thus building data models on the fly. This can be done by one of several methods proposed in the literature [15], [17]. Among these methods, *entropy*, a typical measure of information, emerges as an effective and efficient solution. Hence, we use entropy to represent the anomalous and normal behaviors of nodes and then compare them. More specifically, assume:

- $n$  reporting nodes;
- $p_i$  is the probability that node  $i$  is an attacker (if all nodes are well-behaving, then they can be attackers with the same probability);

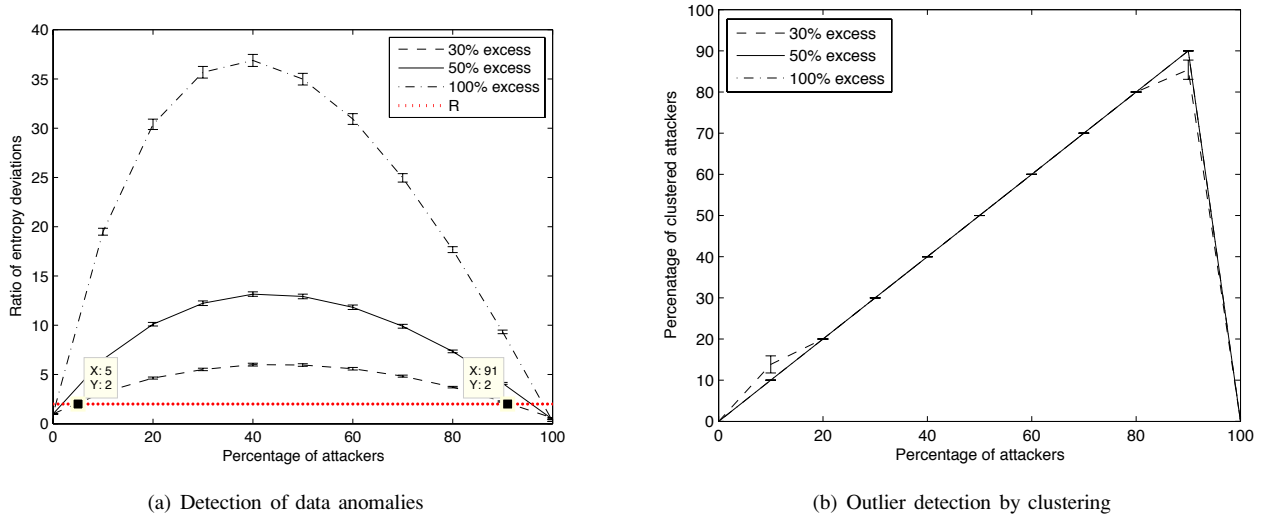


Fig. 4. Performance of the MDS in the case of data anomalies.

- $P_0$  is the probability density function of  $p_i$  when there are no attackers (i.e., it represents normal behavior);
- $P_\epsilon$  is the probability density function of  $p_i$  when there are no attackers and the MDS tolerates mild faults up to a given error margin  $\epsilon$ .  $\epsilon$  defines the rate of false positives and false negatives of the MDS.

The behavior of nodes can be represented by the entropy  $H = -\sum_{i=1}^n p_i \log_2 p_i$ . The MDS detects an attack (i.e., an anomaly) in the system if the corresponding attacker probability density function has the following property:

$$\frac{D(P\|P_0)}{D(P_\epsilon\|P_0)} > R \quad (1)$$

where  $D(P\|Q)$  is the *Kullback-Leibler distance* [18] between the two probability density functions  $P$  and  $Q$ ;  $R$  is the detection threshold for anomalous distance ratios.

As entropy alone only reflects the state of the system without identifying the attackers, we use an *outlier detection* algorithm [19] to single out the attackers. Among the possible options, we use the *K-means clustering* algorithm [20] that iteratively and efficiently converges to the number of clusters that minimizes the sum of distances of all points to the corresponding cluster centroids. Based on the above, the operation of the MDS can be summarized as shown in Fig. 3.

A VN application determines the corresponding known misbehaviors, the sample size  $n$ , the algorithm for computing the  $p_i$  values, the reference probability density functions  $P_0$  and  $P_\epsilon$ , and the detection threshold  $R$ ; the choice of  $\epsilon$  and  $R$  allows tuning the rates of false positives and false negatives. We describe in detail an example application of the proposed MDS below.

It should be noted that the MDS detects only nodes that are in its current neighborhood, based on their locations and timestamps. This limits the load of attacker detection to the attacker's neighbors. This way, the MDS does not distinguish between data originators and data relays. In fact, a (relay) node that propagates false data constitutes a vulnerable point and needs to be contained.

#### A. Example Application of the MDS

In a typical VN application, a vehicle stopped on the roadside emits warnings to alert other vehicles of an event (e.g., an accident) that requires their action (e.g., to slow down). These warnings (containing the location of the event) are destined to all vehicles within a 1 km range. As the typical range of DSRC is 300 m, intermediate vehicles need to forward the warnings over several hops after verifying and signing them (to prevent the uncontrolled propagation of false information). We note that the position of the observing vehicle does not change significantly during the execution time of the MDS because all warnings are transmitted over the high-speed DSRC and all computations are efficient<sup>2</sup> and local.

We assume the attackers attempt, in their warnings, to report the event at a position further from the warning recipients than the actual one; this would prevent vehicles from braking in time, thus causing accidents. Reporting a position closer than the actual one would not cause accidents as vehicles would brake anyway. We consider the case where the observing vehicle is several hops away from the event but within the 1 km area covered by the warnings.

First, the observing vehicle computes the  $p_i$  values corresponding to reporting vehicles as follows. Assuming that the observing vehicle is at location  $(x_o, y_o)$  and that vehicle  $i$  reports that the event/accident is at location  $(x_e, y_e)_i$ , the resulting computed distance from the event is  $d_i = d[(x_e, y_e)_i, (x_o, y_o)]$  where  $d$  denotes a distance function (e.g., Euclidean distance). Let  $q_i$  be the ratio of  $d_i$  to the average of the lower 50% of the reported distances (recall that attackers only enlarge distances and we assume an honest majority). Finally,  $p_i = \frac{q_i}{\sum_{i=1}^n q_i}$  is the probability that vehicle  $i$  is an attacker. The MDS then applies the detection rule in Equation 1. The reference probability density functions  $P_0$  and  $P_\epsilon$  are computed based on the sample size  $n$  and the tolerable error margin  $\epsilon$ . In this example,  $n$  is equal to the

<sup>2</sup>Assuming  $n$  neighbors, the approximation of entropy is upper-bounded by  $O(n^{o(1)} \log n)$  [21]; the complexity of K-means is  $O(n)$  [20].

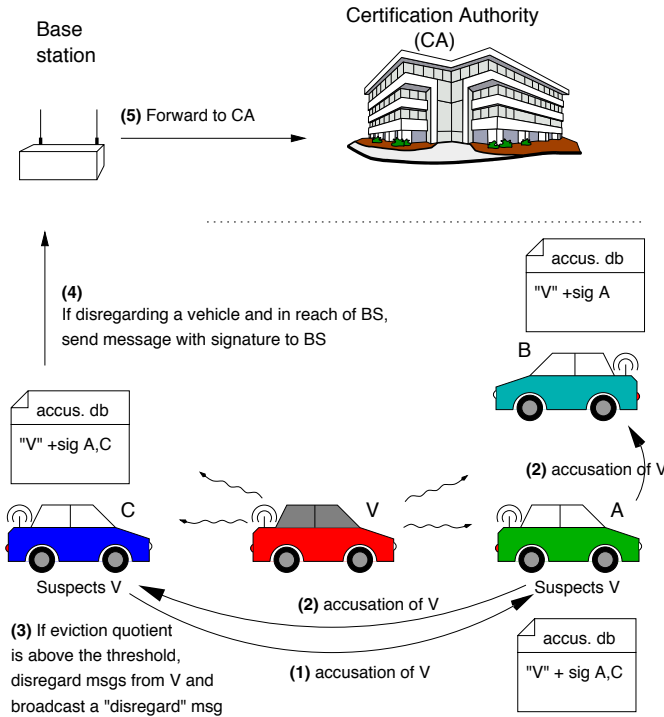


Fig. 5. The LEAVE protocol. Vehicle C has reached the eviction threshold for vehicle V, and broadcasts a disregard message. B is only in the transmission range of A, and gets the information from A when A reaches the threshold and sends a disregard message.

number of reports (about the specific event) received from distinct neighbors.

Fig. 4(a) shows, in the case of 100 nodes and an error margin of 10%, the performance of the MDS (represented by the ratio defined in Equation 1) corresponding to different percentages of attackers. Anomalous enlargement, by the reporting vehicles, of the distance to the event is considered as an attack. The detection threshold  $R$  (i.e., deviation ratio) is set to 2 in this example; we selected this value empirically to reduce false positives and false negatives. We can see that the MDS can detect attacks, for a 30% enlargement, if the percentage of attackers is between 5% and 91%.<sup>3</sup> Attacker percentages larger than 91% cannot be detected because they strongly influence the reference values and thus define the normal, although wrong, behavior. Fig. 4(b) shows the percentage of attackers detected by K-means clustering. We can notice that the larger the percentage of excess (i.e., distance enlargement), the easier it is to detect attackers. This is rather intuitive and acceptable because mild abuse can in fact be a small detection system error due to the highly dynamic environment.

## VI. THE LEAVE PROTOCOL

As mentioned in Section III, being *warned* of the misbehaving nodes allows the observing vehicle to ignore any messages sent by these nodes. Warnings can be triggered by the standalone MDSs running on each vehicle. This is the key

<sup>3</sup>We can notice that the assumption of an honest majority is not necessary in this particular example. The reason is that, in this case, the criterion of an attack is distance enlargement and not deviation from the majority.

concept behind LEAVE, illustrated in Fig. 5. More precisely, vehicles that detect an attacker begin broadcasting *warning* messages to all vehicles in range. The latter can use this information as input to their respective MDSs. In this paper, we consider the case of vehicles that receive warning messages before they are able to make any observations of the attacker and thus rely entirely on these messages. The final step is to report the attackers or faulty nodes to the CA as soon as possible (i.e., when in reach of a base station or mobile unit as defined in Sec. II).

### A. Neighbor Warning System for LEAVE

The warning system relies on the collective information gathered from a vehicle's neighborhood. As all vehicles can be attackers with the same probability, the warning messages may contain correct or wrong *accusations*. Given the limited amount of available evidence, vehicles rely on the assumption of honest majority and crosscheck all received accusations. In this paper, we use a simple algorithm, explained in detail in Appendix II.A, for summing accusations, with an additional feature inspired by [22]: An accusation issued by a node has a lower weight when this node is already accused by other participants. If the sum of weighted accusations (the *eviction quotient*) against a vehicle exceeds a defined threshold, it is locally evicted by LEAVE. More precisely, warning messages are transformed into *disregard* messages that instruct all the neighbors of the attacker to ignore its messages.

It should be stressed here that this algorithm is only an example and other accusation aggregation systems can be devised. We chose this rather simple system because it requires no setup overhead, as incentive systems do [23], nor long observation periods needed by reputation systems [24]. In addition, it involves no interactive mechanisms, thus preventing our system from being dependent on specific participants. As stated earlier, the only requirement for the proposed neighbor warning system to be effective is the existence of an honest majority.

The difference between warning and disregard messages is that a specific number of *supporting signatures* is included by the sender in the disregard message, compared to only one signature in the warning message. This increases the credibility of the message, assuming an honest majority, while maximizing channel efficiency by message aggregation [25].

A vehicle that accumulates enough accusations against an attacker to reach the *eviction threshold* (its computation is detailed in Appendix II.A) is called hereafter a *warned vehicle*. An *initially warned vehicle* is a warned vehicle that disregards all bogus messages from the attacker. This can happen if the vehicle has already reached the warned state before receiving the first message from the attacker (because it has received enough accusations to reach the eviction threshold).

It should be noted that, although LEAVE relies on accusations by nodes, it is highly resilient to attackers as shown in Appendix II.B.

### B. Definition of the Attacker's Neighborhood

The neighbor warning system, introduced in the previous section, relies on the neighbors of a suspected vehicle (*suspect*) to accuse it in case of misbehavior and warn other

vehicles. Once these vehicles have enough information about the suspect, they can evaluate whether it is misbehaving; we refer to them as *evaluators*. Hence, it is important to define the suspect's *neighborhood*  $N$ . New vehicles coming into the communication range of a suspected vehicle will use the information provided by their neighbors to detect misbehavior. It is essential here to avoid the problem whereby a vehicle that is not in the suspect's range, and hence cannot evaluate it, is considered when making a decision (Fig. 6). Therefore, the neighborhood  $N$  of a suspected vehicle also depends on the vehicle evaluating it. We define  $N(s, e)$ , where  $s$  and  $e$  refer to suspect and evaluator, respectively, as the intersection of the coverage areas (defined by the transmission range) of both the suspected vehicle and the vehicle evaluating it. This way, we ensure that all vehicles in  $N$  can receive messages from the suspect, and thus potentially accuse it, while being able to report their accusations to the evaluators.

It should be noted that an evaluator vehicle can select the elements of  $N$  because all vehicles broadcast their position information. The suspect's position is reported in the warning messages received by the evaluator.

## VII. PERFORMANCE EVALUATION

In this section, we evaluate the performance of LEAVE under stringent VN conditions. The following results show that eviction of misbehaving nodes can be done efficiently. A detailed performance analysis of the usage of Bloom filters can be found in Appendix I.

As LEAVE relies on the ad hoc operation of vehicles within short time delays, we simulate it using *ns-2* with the MAC-layer parameters of IEEE 802.11p. We consider three different parameters in our evaluations. The first one is the traffic model: we used a freeway (FW), a city (West University, or WU), and a mixed (freeway/city) (Afton Oak, or AO) scenarios.<sup>4</sup> The second and third factors are the density of vehicles and their average speed, respectively. The presented results are the average of 50 simulation runs. As the MDS runs locally on each vehicle, its computational delay is small and hence not critical for the system operation; in the simulations, we do not include this delay. We place one adversary in the system, but the same results apply to multiple adversaries, assuming there is an honest majority (LEAVE runs separately for each adversary).

### A. Vehicle Density

In the WU and AO scenarios, for very low vehicle densities the percentage of *initially warned vehicles* (Section VI-A) is low (Fig. 7(a)). In this case, it is not possible to pass information from one vehicle to another in a reliable way and thus warn other vehicles before they encounter the adversary. However, this percentage grows with increasing density and stabilizes between 80% and 90%. Vehicles not initially warned need less than 2 s to begin disregarding messages at high densities (Fig. 7(b)). For the freeway scenario, there is a slight decrease in performance at very high densities. This may be

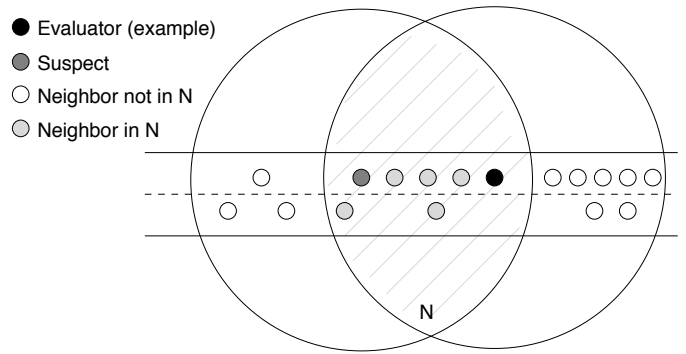


Fig. 6. Definition of the attacker's neighborhood.

explained by the fact that the number of packet collisions increases when the density increases.

Overall, even with a relatively low density of VN-enabled vehicles, which will be the case with a low market penetration at the beginning of VN deployment, LEAVE is still able to accumulate enough information to perform successfully within the constraints imposed by short contact times (e.g., assuming a transmission range of 300m and two vehicles moving in opposite directions on the same highway, each at an average speed of 100 km/h, their contact time is merely around 11 s).

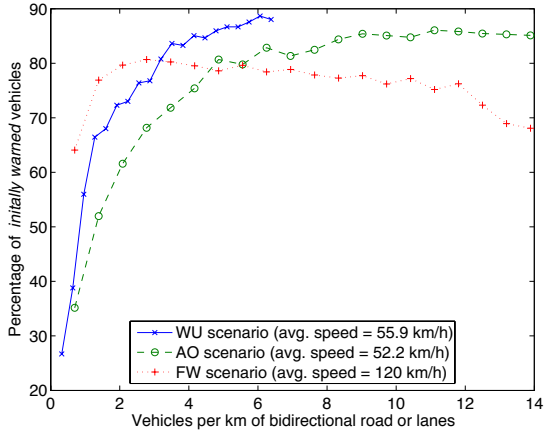
### B. Average Speed

We evaluate the same metrics for different average speeds in the three scenarios (Figs. 7(c) and 7(d)). In the urban areas (AO and WU), we can see that higher speeds give better results: again, because more participants can be contacted in a given time interval, it is possible to accumulate more accusations against the attacker and to warn other vehicles (in these two cases, the maximum average speed is 90 km/h, higher values are rare). In the freeway scenario, the average speed is much higher, and performance decreases slightly for very high speeds: this can be explained by the fact that contact times becomes very short. In this case, some messages may not be received, thus resulting in a slightly higher time to reach the necessary threshold. Still, we can see that LEAVE operates within acceptable delays and covers a considerable percentage of concerned vehicles.

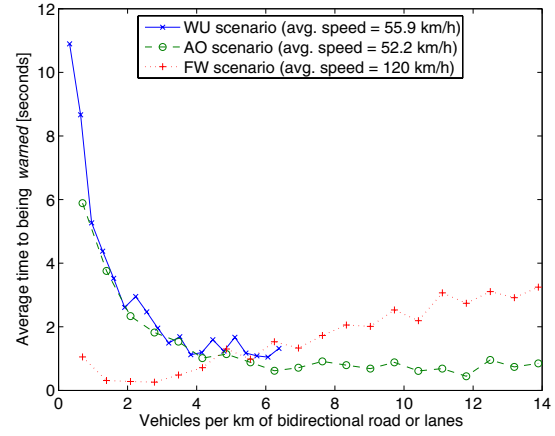
### C. Effect of Warning Rebroadcast Interval

If a vehicle continues to receive bogus messages from the attacker, it does not send warning messages continuously. Rather, it repeats them once per a *Warning Rebroadcast Interval* (WRI), as long as the attack persists. The WRI is used to prevent vehicles from flooding the channel with accusation messages, thus preventing DoS attacks based on excessive channel load or on computation overhead (due to digitally signing warning messages). However, if a vehicle sends a warning message only once and then stops participating in the warning process against a potential attacker, newly arriving vehicles will not be able to accumulate enough accusations to disregard the attacker's messages. Hence, the parameter WRI is a tradeoff between overhead (sending too often the same warnings) and responsiveness to attackers (quickly informing new neighbors about misbehaving vehicles).

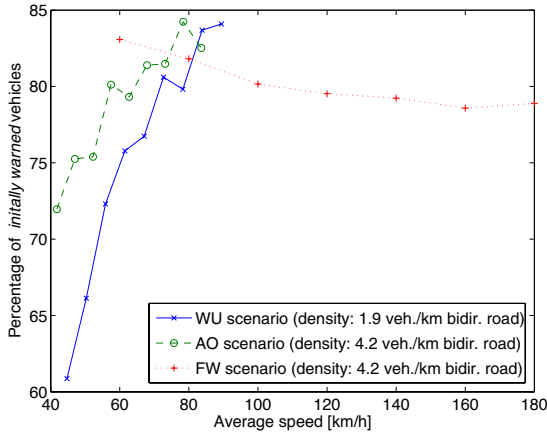
<sup>4</sup>WU and AO are realistic scenarios taken from [26]. The framework used for developing the simulations in this section is available at [27].



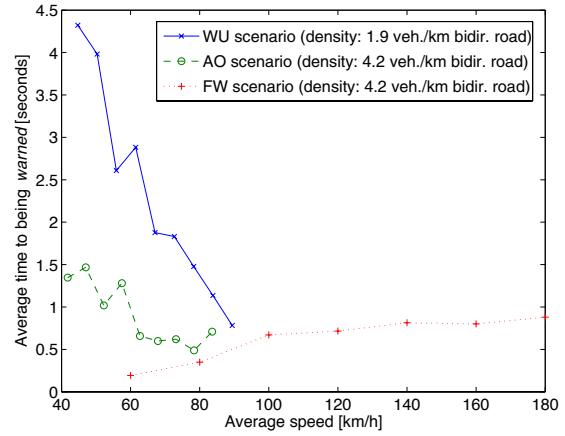
(a) Percentage of initially warned vehicles vs. vehicle density (in the FW scenario, the density is per lane)



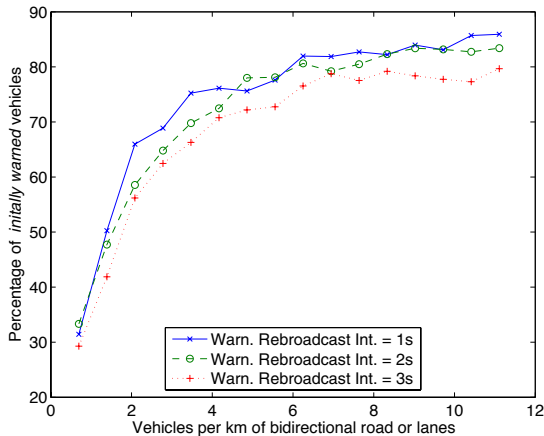
(b) Average time to being warned vs. vehicle density



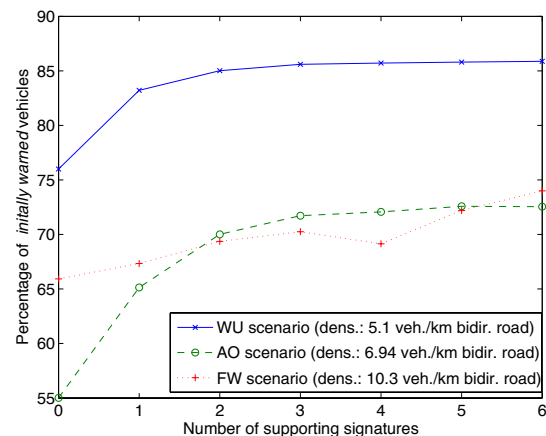
(c) Percentage of initially warned vehicles vs. average vehicle speed



(d) Average time to being warned vs. average vehicle speed



(e) Effect of the WRI parameter (AO scenario)



(f) Effect of the number of supporting signatures

Fig. 7. System performance vs. vehicle density, average speed, WRI, and number of supporting signatures.

Changing the WRI has only a small effect on the percentage of initially warned nodes in the case of low vehicle density (Fig. 7(e)). In higher density situations we can observe a slight increase of warned vehicles with smaller WRIs (i.e., when sending accusations more frequently), but this also increases channel load. As the results show, WRI can be set to a

high value (and thus incur low overhead) without significant degradation in the performance of LEAVE.

#### D. Number of Supporting Signatures

The *number of supporting signatures* (Section VI-A) is a parameter of LEAVE. It may be influenced by the signature



size of the cryptographic system in use. But a minimum number of signatures is required to ensure the credibility of disregard messages. Fig. 7(f) shows that the number of warned vehicles changes considerably with low numbers of supporting signatures. But starting with only 4 signatures, there is little change in the results. This can be explained by the fact that, starting from this point, the redundancy of supporting signatures increases.

To cope with the abuse of disregard messages, a vehicle should still crosscheck several disregard messages and verify that the total number of supporting signatures received in these messages is larger than the majority of nodes in the neighborhood  $N$  (assuming an honest majority). The advantage of relying on small rather than large disregard messages is smaller retransmission overhead if a collision happens on the wireless channel.

### VIII. RELATED WORK

Revocation has been considered mostly in the context of the wireline Internet and the design of Public Key Infrastructure (PKI) services [9]. Nevertheless, the design of mechanisms to disseminate the revocation information across systems similar to VNs has not been considered in the wireline Internet context (for a survey and discussion of tradeoffs see [28], [29]). Due to the network volatility and scale, the overhead of querying a server to obtain timely revocation status, assuming the server is reachable, could be impractically high. For the same reasons, schemes that distribute the load of a server to a set of participating clients [30] (to redundantly forward revocation information) would not be practical for deployment within the VN, but only meaningful behind the fixed infrastructure.

Existing works on VN security [2], [8], [10], [31] propose the use of a PKI and digital signatures but do not provide any mechanisms for certificate revocation, even though it is a required component of any PKI-based solution. Different aspects of revocation were discussed in [7], [32], [33] without a complete solution provided. In the context of VNs, the IEEE 1609.2 Draft Standard [4] is the only reference on certificate revocation. It proposes the distribution of CRLs and short-lived certificates, but does not elaborate how to achieve this. Short-lived certificates are also proposed in [6]. Short lifetimes are essentially a means of revocation that achieves efficiency but opens a vulnerability window; such an approach is not appropriate for a life-critical VN environment.<sup>5</sup> Moreover, certificates have to be refreshed frequently to keep the vulnerability window very small. This could create high loads both on the CA and the network.

The literature on VNs already contains methods for adversary detection. For example, threshold-based tests to verify positioning information in VNs were proposed in [16]. In [13], a more general framework for malicious data detection compares the received data to a *model of the VN*; but the paper provides no details on possible tests.

The application of information-theoretic measures to anomaly detection was previously studied in the literature [34],

[35], [36], but mainly in the context of the wired Internet. Most notably, [36] successfully applied the notion of relative entropy (also known as the Kullback-Leibler distance) to measure the similarity between two datasets.

Instantiating a CA in the context of mobile ad hoc networks was investigated, with the distribution of its functionality to a number of servers [37]. However, this scheme does not consider the problem of revocation, especially in a highly mobile environment like a VN. Instantiation of the CA functionality (or part thereof) by impromptu coalitions of network nodes (e.g., [22], [38]) cannot be applicable in VN systems. Allowing any ad hoc and, in general, small subset of adversarial nodes to maliciously accuse and revoke legitimate nodes would be an unacceptable breach of the VN system security where accountability and liability are mandatory.

### IX. CONCLUSION

In this paper, we propose a framework to thwart internal attackers in vehicular networks. The eviction of faulty or attacking nodes is crucial to the robustness of vehicular communication systems. As revocation is the primary means to achieve this, we designed two protocols tailored to the characteristics of the VN environment. To eliminate the vulnerability window, due to the latency for the authority to identify faulty or misbehaving nodes and distribute revocation information, we designed a scheme that can *robustly* and *efficiently* achieve their isolation, as well as contribute to their eventual revocation. This is done with the help of a misbehavior detection module and a distributed eviction protocol. Given the broad scope of the subject tackled in this paper, there is ample space for future work on each of the individual components of our framework. Nonetheless, our results evaluating the instantiation proposed in this paper show that our scheme is practical, efficient, and effective in isolating misbehaving and faulty nodes.

### ACKNOWLEDGEMENTS

We would like to thank Paul Drielsma, Virgil D. Gligor, Marcin Poturalski, Patrick Schaller and the anonymous reviewers for their helpful feedback on earlier versions of this work.

### APPENDIX I: BLOOM FILTERS

A Bloom filter, illustrated in Fig. 8, consists of a  $m$ -bit vector with all its bits initially set to zero. An element (a public key in our context) can be included in the filter by (i) hashing it with  $k$  independent hash functions that output numbers in the range  $1, \dots, m$ , and (ii) setting to 1 the vector bit each hash function points to. It is possible that one bit is set to 1 multiple times due to the addition of several elements. To check if a given element is contained in the filter, the element is hashed and the corresponding filter bits are checked: If at least one of those bits is zero, the element is not contained in the filter. Otherwise, if all necessary  $k$  bits are set, the element is included with high probability. The corresponding bit could have been set also due to multiple additions of other elements. The more elements added, the larger the probability of false positives. In the context of revocation by C<sup>2</sup>RLs, the

<sup>5</sup>An exception can be context-specific credentials, allocated, for example, to a vehicle entering a highway segment and “purchasing” access to a service. However, this is orthogonal to the problem we are considering here.

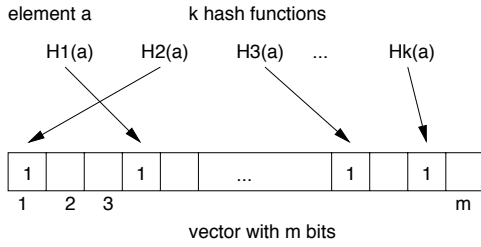


Fig. 8. Bloom filter concept.

nodes validate the certificates included in received messages by checking the Bloom filter. We discuss quantitative aspects of Bloom filters in the next section.

### A. Bloom Filter Performance

Bloom filters provide a compression tool with configurable compression gain ( $c$ ) and false positives rate ( $p_{fp}$ ). The configurable parameters of the filter are:

- The filter vector size  $m$ .
- The number of hash functions  $k$ .

An additional input, however not configurable, is the number of list entries (certificate IDs)  $L$ , each considered to be  $l$  bits long.

A large value of  $m$  considerably reduces the false positives rate  $p_{fp} = (1 - (1 - 1/m)^{kL})^k \approx (1 - e^{-kL/m})^k$ , at the cost of decreasing the compression gain  $c = L \times l/m$ . The choice of  $m$  can be derived from Fig. 9(a), where the number of hash functions  $k$  is chosen to be optimal ( $\ln(2) \times m/n$ , when  $dp_{fp}/dk = 0$ ). Taking also into consideration the range of the number of list entries  $L$ , we choose  $m = 20$  KBytes, transmittable over the considered radio channels within short time limits (e.g., around 27 ms over a 6 Mbps DRSC channel).

The choice of the optimal number of hash functions  $k$  improves the efficiency, at the cost of increased system complexity. In fact, to use a variable number of hash functions, the CA must transmit the used value of  $k$  together with the filter. At the receiving side, the verifier must learn  $k$  and use  $k$  entries of a pre-established list of hash functions.

To avoid this complexity, we use a fixed number of hash functions  $k$ . Fig. 9(b) shows the case where  $k = 10$  (for  $m = 20$  KBytes), a compromise between computation complexity and false positives rate. We can see that the resulting false positives rate is reasonably low for small  $n$  and converges to the performance provided by optimal values of  $k$  when  $n$  increases. As the for the considerably high compression gain  $c$  ( $10 < c < 138$ ), it is independent of the number of hash functions  $k$  (whether fixed or variable/optimal).

## APPENDIX II: DETAILS OF LEAVE

### B. Computation of the Eviction Quotient

An observing vehicle uses the following parameters to calculate the eviction quotient for vehicle  $j$  (with accusations from different vehicles  $i$ ):

- $A_i$  is the total number of accusations (issued by different vehicles) against vehicle  $i$ .  $A_i$  is used to reduce the weight (i.e., credibility) of the accusations made by vehicle  $i$  if it was already accused.

- $P_i$  is the accumulated sum of  $|N_i|$ , the number of  $i$ 's neighbors (as explained in Sec. VI-B) that the observing vehicle has encountered.
- $\alpha_i$  is the normalized value of the total number of accusations with respect to the total number of neighbors of vehicle  $i$  ( $0 \leq \alpha_i \leq 1$ ). This value is computed as follows:  $\alpha_i = \frac{A_i}{P_i}$ .
- $\omega_i$  is the weight of any accusation made by vehicle  $i$ . This weight depends on the number of accusations made against  $i$ :  $\omega_i = 1 - \alpha_i$ , giving  $0 \leq \omega_i \leq 1$ . Therefore, the weight of a node against which there are no accusations equals 1.
- $Q_j$  is the eviction quotient defining whether vehicle  $j$  should be evicted. It is computed as follows:  $Q_j = \frac{1}{P_j} (\sum_{i=1}^{P_j} \sigma_{ij} \omega_i)$ , where  $\sigma_{ij} = 1$  if there is an accusation against  $j$  issued by  $i$ , and  $\sigma_{ij} = 0$  otherwise.

The eviction quotient threshold ( $Q_T$ ) is a configurable parameter. A typical value would be 0.5 (majority voting). If  $Q_j > Q_T$ , vehicle  $j$ 's messages are disregarded.

### C. Resilience to Attackers

As LEAVE relies on nodes accusing attackers, there is a potential for abuse by attackers: A group of colluding attackers can accuse honest nodes and cause their eviction. In this section, we analyze LEAVE's resilience to such attacks. Assuming all nodes have roughly the same number of neighbors,  $P_j \approx P$ , the eviction quotient becomes  $Q_j = \frac{1}{P} \sum_{i=1}^P \sigma_{ij} \omega_i$ . Let  $x$  be the fraction of attackers in the neighborhood. We can distinguish two cases:

**Case 1:** Attackers accuse all honest nodes; honest nodes do not accuse attackers. We consider this to be a strong attacker case. The eviction quotient required for a successful attack is:

$$Q = \frac{1}{P} \sum_{i=1}^{xP} 1 = x \quad (2)$$

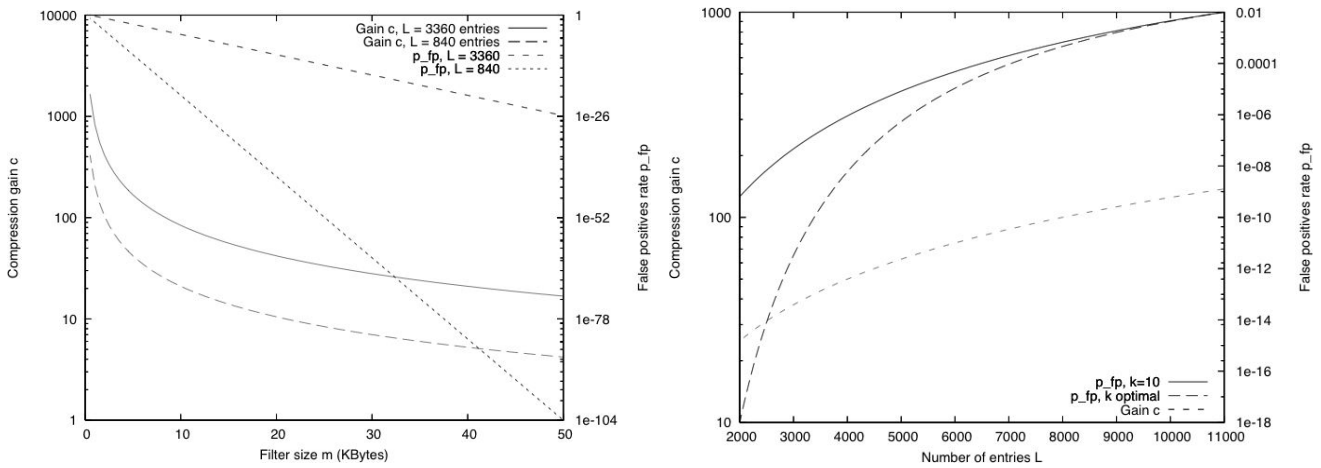
**Case 2:** Attackers accuse all honest nodes; honest nodes accuse all attackers. We consider this to be a weak attacker case. The eviction quotient required for a successful attack is:

$$Q = \frac{1}{P} \sum_{i=1}^{xP} (1 - \frac{P - xP}{P}) = x^2 \quad (3)$$

Fig. 10 shows the required percentage of attackers for a successful attack, given an eviction threshold of 0.5. We can see that, under the assumption of honest majority, LEAVE is highly resilient to colluding accusers.

## REFERENCES

- [1] J. Blum and A. Eskandarian, "The threat of intelligent collisions," *IT Professional*, vol. 6, no. 1, pp. 24–29, Jan.-Feb. 2004.
- [2] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proceedings of HotNets-IV*, 2005.
- [3] P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "Securing vehicular communications - assumptions, requirements, and principles," in *Workshop on Embedded Security in Cars (escar'06)*, 2006.
- [4] "IEEE P1609.2 Version 1 - Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages," *In development*, 2006.
- [5] M. Gerlach, "VaneSe - An approach to VANET security," in *V2VCOM*, 2005.



(a) Compression gain and false positives rate vs. filter size, using optimal  $k$  (b) Compression gain and false positives rate vs. number of entries  $L$ , using  $k=10$  hash functions

Fig. 9. Bloom filter performance.

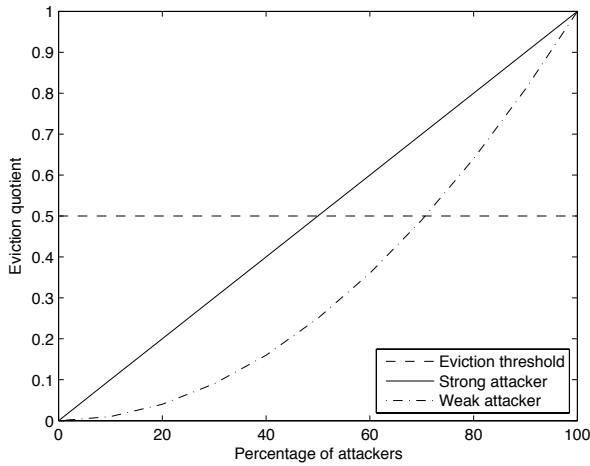


Fig. 10. LEAVE: resilience to attackers.

[6] M. Jakobsson and S. Wetzel, "Efficient attribute authentication with applications to ad hoc networks," in *Proceedings of VANET'04*, 2004.  
 [7] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications Magazine*, vol. 13, no. 5, pp. 8–15, 2006.  
 [8] M. E. Zarki, S. Mehrotra, G. Tsudik, and N. Venkatasubramanian, "Security issues in a future vehicular network," in *European Wireless*, 2002.  
 [9] R. Housley, W. Polk, W. Ford, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 3280, 2002.  
 [10] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of SASN'05*, 2005.  
 [11] "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems – 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ASTM E2213-03, 2003.  
 [12] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov, "Cryptographic processors—A survey," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 357–369, 2006.  
 [13] J. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proceedings of VANET'04*, 2004.  
 [14] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.  
 [15] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless

ad hoc networks," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 48–60, Feb. 2004.  
 [16] T. Leinmüller, C. Maihöfer, E. Schoch, and F. Kargl, "Improved security in geographic ad hoc routing through autonomous position verification," in *Proceedings of VANET'06*, 2006.  
 [17] G. Carl, G. Kesidis, R. Brooks, and S. Rai, "Denial-of-Service attack-detection techniques," *IEEE Internet Computing*, vol. 10, no. 1, pp. 82–89, Jan.-Feb. 2006.  
 [18] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951.  
 [19] P. Rousseeuw and A. Leroy, *Robust regression and outlier detection*. New York, NY, USA: John Wiley & Sons, Inc., 1987.  
 [20] A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.  
 [21] T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld, "The complexity of approximating the entropy," *SIAM Journal on Computing*, vol. 35, no. 1, pp. 132–150, 2005.  
 [22] C. Crépeau and C. Davis, "A certificate revocation scheme for wireless ad hoc networks," in *Proceedings of SASN'03*, 2003.  
 [23] S. Zhong, J. Chen, and Y. Yang, "Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proceedings of Infocom'03*, 2003.  
 [24] S. Buchegger and J.-Y. L. Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," in *Proceedings of P2PEcon'04*, 2004.  
 [25] M. Raya, A. Aziz, and J.-P. Hubaux, "Efficient secure aggregation in VANETs," in *ACM Workshop on Vehicular Ad hoc Networks (VANET)*, 2006.  
 [26] A. Saha and D. Johnson, "Modeling mobility for vehicular ad-hoc networks," in *Proceedings of VANET'04 (Poster session)*, 2004.  
 [27] <http://ivc.epfl.ch>.  
 [28] P. Wohlmacher, "Digital certificates: a survey of revocation methods," in *Proceedings of Multimedia'00*, 2000.  
 [29] P. Zheng, "Tradeoffs in certificate revocation schemes," *SIGCOMM Computing Communication Review*, vol. 33, no. 2, pp. 103–112, 2003.  
 [30] R. Wright, P. Lincoln, and J. Millen, "Efficient fault-tolerant certificate revocation," in *Proceedings of CCS'00*, 2000.  
 [31] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks*, vol. 15, no. 1, pp. 39 – 68, 2007.  
 [32] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya, "Architecture for secure and private vehicular communications," in *Proceedings of ITST'07*, 2007.  
 [33] G. Calandriello, P. Papadimitratos, A. Lloy, and J.-P. Hubaux, "Efficient and robust pseudonymous authentication in vanets," in *Proceedings of VANET'07*, 2007.  
 [34] E. Eiland and L. Liebrock, "An application of information theory to intrusion detection," in *Proceedings of IWIA'06*, 2006.  
 [35] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to DDoS attack detection and response," in *Proceedings of the DARPA Information Survivability Conference and Exposition*, 2003.  
 [36] W. Lee and D. Xiang, "Information-theoretic measures for anomaly

detection,” in *Proceedings of the IEEE Symposium on Security and Privacy*, 2001.

- [37] L. Zhou and Z. Haas, “Securing ad hoc networks,” *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.
- [38] J. Kong, H. Luo, K. Xu, D. Gu, M. Gerla, and S. Lu, “Adaptive security for multilevel ad hoc networks,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 533–547, 2002.



**Maxim Raya** received his B.Eng. degree in Computer and Communications Engineering in 2002 from the American University of Beirut, Lebanon. He is currently pursuing his Ph.D. studies under the supervision of prof. Jean-Pierre Hubaux at EPFL. His research interests are in the area of security in wireless networks, and especially vehicular networks. He served in the program committee of VANET 2007. <http://people.epfl.ch/maxim.raya>.



**Panagiotis Papadimitratos** received his PhD degree in Electrical and Computer Engineering from Cornell University Ithaca, NY, in 2005. He joined then the Department of Electrical and Computer Engineering at Virginia Tech, Blacksburg, VA, as a research associate. Panos is currently a senior researcher with the School of Computer and Communication Sciences at EPFL. His research is concerned with networking protocols, network security, ad hoc and sensor networks, and wireless and mobile systems. He has authored more than 35 technical

publications on these topics. He has served in the technical program committees of numerous conferences and workshops, among which are ACM ASIACCS, ACM WiSec, ACM VANET, and IEEE MASS. He has delivered several invited talks and lectures, including a tutorial on security and cooperation in wireless networks delivered at ACM MobiCom 2007. His personal website is <http://people.epfl.ch/panos.papadimitratos>.



**Imad Aad** is a senior researcher at DoCoMo Euro-Labs, Germany, working within the Future Networking Lab. He got his Electrical and Electronics engineering degree in 1998 from the Lebanese University, Beirut. He got his M.S. degree in 1999 from the University of Nice - Sophia Antipolis, then his Ph.D. degree from Joseph Fourier University, France, in 2003. He prepared his Ph.D. on quality of service in wireless LANs at INRIA, France, within the Planète team. He worked at EPFL within the LCA team as a senior researcher from 2003 to 2005

where he worked on cheating and cheating detection issues, DoS attacks and resilience and on general security aspects in wireless networks. He joined DoCoMo Euro-Labs in December 2005 where he is working on cooperative coding, anonymity, quality of service, beam antennas and overlays in wireless networks. He is co-chairing WiOpt 2008 workshops and served in the program committee of ESAS, WiOpt, ACM MM, WinSys, CFIP, WCNC and IWCMC. <http://imad.aad.name>



**Daniel Jungels** finished his M.S. degree in Communication Systems at EPFL in 2005. Currently, he is a software developer at HITEC Luxembourg S.A., participating in an FP6 framework programme research project about ubiquitous IP networking, including IP mobility and IPv6. He is also working on industrial projects in the domain of traffic planning and supervision on highways.



**Jean-Pierre Hubaux** is a full professor at EPFL. His research activity is focused on wireless networks, with a special interest in security and cooperation issues. He has been strongly involved in the definition and launching phases of a new National Competence Center in Research named “Mobile Information and Communication Systems” (NCCR/MICS), since its genesis in 1999. In this framework, he has made several contributions in the areas of key management, secure positioning, incentives for cooperation, power management, and

group communication in sensor and ad hoc networks. In 2003, he identified the security of vehicular networks as one of the main research challenges for real-world mobile ad hoc networks. In 2007, he completed a graduate textbook entitled “Security and Cooperation in Wireless Networks”, with Levente Buttyan. He is the chairman of the steering committees of ACM Mobihoc 2008 and of ACM WiSec 2008. He is a member of the steering committee of IEEE Transactions on Mobile Computing and an associate editor of Foundations and Trends in Networking. He has been serving on the program committees of numerous conferences and workshops, including SIGCOMM, Infocom, Mobicom, Mobihoc, SenSys, WiSe, and VANET. He is a member of the Federal Communications Commission (ComCom), the “Swiss FCC”. He held visiting positions at the IBM T.J. Watson Research Center and at the University of California at Berkeley. After completing his studies in electrical engineering at Politecnico di Milano, he worked 10 years in France with Alcatel, where he was involved in R&D activities. <http://people.epfl.ch/jean-pierre.hubaux>.