Fast exclusion of errant devices from vehicular networks

Tyler Moore*, Maxim Raya†, Jolyon Clulow*, Panos Papadimitratos†, Ross Anderson*, Jean-Pierre Hubaux†
*Computer Laboratory, University of Cambridge, UK
†LCA Laboratory, EPFL, Switzerland
firstname.lastname@cl.cam.ac.uk, firstname.lastname@epfl.ch

Abstract*—Vehicular networks, in which cars communicate wirelessly to exchange information on traffic conditions, offer a promising way to improve road safety. Yet ensuring the correct functioning of such a system is essential: malicious or faulty devices transmitting inaccurate messages could trigger accidents. Therefore, any errant device, along with the messages it generates, must be identified and ignored as quickly as possible. This task is especially challenging because traditional approaches to revoking credentials use a central authority, causing long delays during which the network is vulnerable. To eliminate this window of vulnerability, we propose that vehicles locally decide whether to exclude errant devices. We describe two ways of doing so: first, LEAVE, an existing protocol which allows devices to vote by exchanging signed claims of impropriety, and second, Stinger, a new protocol where a device unilaterally removes a misbehaving neighbor by agreeing to limit its own participation. We provide detailed simulations that offer insight into the protocols' operations in the context of vehicular networks and enable a powerful comparison between the strategies. We compare the security and performance properties of LEAVE and Stinger while varying attacker capabilities, traffic conditions, and the accuracy of the misbehavior detection mechanisms. We identify several interesting trade-offs: Stinger is significantly faster than LEAVE at removing errant devices, but LEAVE excludes fewer good devices when the attacker has compromised several devices simultaneously; LEAVE is better at handling false positives, but Stinger scales better when the traffic density increases. As a result, we conclude by outlining a combined protocol that balances the security and performance characteristics of both strategies.

I. Introduction

Recently, consortia of automobile manufacturers in the US [19], Europe [4] and Japan [1] have begun investigating ways to equip vehicles with wireless radios for communicating safety information. For instance, cars might send each other collision warnings or traffic congestion notifications.

Ensuring the integrity of safety communications is paramount. Compromised transmitters might send bogus information for reasons that are selfish (e.g., pretending there is an automobile accident to divert traffic away from the chosen path and enjoy an uncongested ride) or malicious (e.g., faking location information to encourage collisions). Alternatively, the transmitters may simply be broken, which is a less sinister but entirely plausible threat to message integrity.

Whenever a device starts sending bad information, the long-term solution is for the certification authority (e.g., the Department of Motor Vehicles) to revoke the credentials of the offending device. However, this process takes time, from the collection of evidence to the resolution of disputed claims. In the interim, ongoing attacks could endanger passenger safety. Thus, there is a need to rapidly isolate such errant devices and prevent them from spreading incorrect data. One solution is for the cars observing misbehavior to temporarily exclude the responsible bad device until the certification authority is notified and takes appropriate action. In this paper, we consider ways to reach such a local decision while at the same time maximizing efficiency and security.

We first describe an already proposed local decision mechanism, called LEAVE, where nodes vote to exclude errant devices by exchanging signed claims of impropriety [16]. We then propose a new protocol, called Stinger, in which a node can unilaterally remove a perceived misbehaving neighbor by limiting its own participation. Stinger is a tempered adaptation of the suicide protocol proposed for ad-hoc networks in [11].

We then set out to contrast the LEAVE and Stinger mechanisms, finding that they often exhibit complementary security and performance properties. This comparison is done through extensive simulations and involves a detailed comparative framework which supports varying attacker capabilities, characteristics of detection mechanisms, and traffic conditions. We demonstrate the circumstances under which voting-based LEAVE and unilateral Stinger perform best. Finally, we describe a hybrid protocol where cars are free to dynamically choose between LEAVE and Stinger as their eviction strategy depending upon traffic conditions. This adaptive strategy promises to achieve a more favorable balance of security and performance than either strategy alone.

II. Vehicular networks

We now describe the operational characteristics of vehicular networks.

A. System model

Existing automotive authorities are likely to become certification authorities (CAs). Each would be responsible for the identity management of all vehicles registered in its respective geographic region. Vehicles register with exactly one CA. Each node has a unique identity, a pair of private and public cryptographic keys, and a certificate issued by the CA.

Messages are transmitted periodically, e.g., every 0.3 s for safety messages, or triggered by in-vehicle or network events. Most traffic is broadcast to limited regions of the network. All

safety-related messages include the time and geographical coordinates of the sender, in addition to other application-specific information. Each message is also signed and accompanied by the sender's certificate. It is widely accepted that asymmetric cryptography is feasible for vehicular networks [15].

Safety messages may need to propagate across multiple hops. In this case, they are signed and include the coordinates and timestamp of the last relaying node, along with the originator's signature, coordinates and timestamp. This chain of signatures helps ensure the freshness of the information while limiting the propagation of illegitimate information. A received safety message is discarded if the difference between its timestamp and the timestamp of the receiver is greater than a system-specific constant accounting for clock drift, propagation and processing delays. Moreover, a message is discarded (by a receiver) if the coordinates of its sender/relay indicate that the receiver is outside the sender's maximum nominal wireless communication range. These validations are applied at each hop.

At the data link layer, the Dedicated Short Range Communications (DSRC) protocol [2], currently being standardized as IEEE 802.11p, provides transmission ranges of typically 300 to 1000 m, with data rates in the 6-27 Mbps range. Beyond DSRC, vehicular networks could also leverage other wireless communication technologies. In this paper, we assume that 802.11p is used.

A subset of network nodes form the infrastructure, comprised of the short-range DSRC base stations and mobile units. The latter include public safety vehicles (e.g., highway assistance and fire-fighting vehicles), police vehicles, and public transport vehicles (e.g., buses, trams). Infrastructure nodes serve as the gateway of the CA to and from the vehicular network; the connection of the CA to the static infrastructure nodes is over wired secure links. However, we do not assume that the CA must be accessible from the vehicular network at all times. The LEAVE and Stinger exclusion[1] mechanisms are carried out by ordinary vehicles, not infrastructure nodes.

B. Threat model

An adversary, or attacker, may control a number of nodes that deviate from the legitimate vehicular network protocols. Nodes can also be faulty due to equipment failures. A detailed discussion of adversary and fault models is given in [13]. As our proposed mechanisms apply to both misbehaving and errant devices, we use both terms interchangeably. We emphasize that we are concerned with misbehaving nodes equipped with valid credentials.

We consider two types of attacker strategy. *False information dissemination* may be a very effective attack, when compared to deviations from networking protocols. The motivation for false information dissemination attacks may be malicious (e.g., sending fake braking information to trigger an accident) or selfish (e.g., claiming an accident has occurred to clear congestion). The adversary could either manipulate the sensory inputs or compromise the protocol stack and the computing platform [13]. An attacker may also control incoming communication, e.g., selectively erasing messages received by its on-board platform.

A second attacker strategy is *exclusion mechanism abuse*. In the next section, we discuss two proposals for excluding bad devices from participating on the network. These strategies, along with any other mechanism that attempts to exclude bad devices, may be abused by an adversary trying to remove good devices instead of bad ones.

[1]We use the terms eviction and exclusion interchangeably.

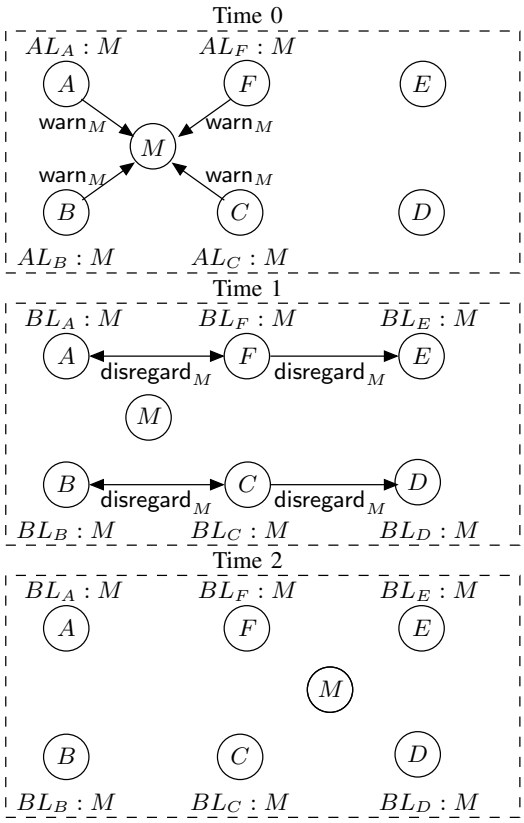

Fig. 1. The LEAVE protocol. Vehicles A, B, C, and F accuse vehicle M and put it in their respective accusation lists AL. Once the exclusion threshold is reached, disregard messages are broadcast. M is then added to the blacklist BL of all accusing vehicles plus the vehicles receiving disregard messages (D and E). At time 0, D and E are not in M's transmission range and cannot detect its misbehavior.

III. Excluding errant devices

In this section we describe the LEAVE and Stinger mechanisms.

A. LEAVE

LEAVE (Local Eviction of Attackers by Voting Evaluators) [16] is illustrated in Figure 1. Vehicles detecting an errant device broadcast *warning* messages to all vehicles in range. Any vehicle receiving a warning message adds the warned device to an *accusation list*. Once enough warning votes against a node are collected, its identifier is added to a local blacklist. After nodes are added to the blacklists,

additional *disregard* messages are repeatedly broadcast to the local neighborhood instructing the receiving nodes to ignore the attacker's messages. Hence, vehicles using LEAVE can be made aware of bad vehicles before interacting with them. Finally, the evicted nodes are reported to the CA once within reach of an infrastructure node.

Deciding when to warn a node using LEAVE is actually more subtle than surpassing a simple numerical threshold of warning votes. Rather, it is based on exceeding an *exclusion quotient*, a sum of weighted accusations relative to the size of a vehicle's neighborhood (the LEAVE paper used an exclusion quotient of 0.5). The exclusion quotient discounts accusations from users who have themselves been accused by others, as proposed by Crépeau and Davis in [5]. For disregard messages, a simple threshold is used (the LEAVE paper used a threshold of 4 votes). To demonstrate their legitimacy, disregard messages include supporting signatures from this threshold of users.

To be secure, LEAVE requires an *honest majority*: every good node must always have more good neighbors than bad. If the attacker controls more devices than the threshold required to send disregard messages, then bad devices can eject any good device at will.

B. Stinger

In [11], Moore *et al.* propose several strategies that enable nodes to remove compromised devices from an ad-hoc network. We now discuss how one strategy presented – *suicide attacks* – can be adapted for use in vehicular networks.

Procedures for removing a bad device are much simpler when taken by a single node. Should a node believe another has misbehaved, it can unilaterally remove the offender. Of course, a malicious node could falsely accuse legitimate ones. Therefore, the act of punishment must be made costly for the deciding node. Suicide attacks remove both the accused *and* accuser from the network. Upon detecting a node M engaging in some illegal activity, node A sends a *suicide note* $\mathsf{suicide}_{A,M}$ with the identities of both A and M. The other nodes now disregard both A and M. Sacrificing future participation is so costly that it unequivocally demonstrates the veracity of the node's claim.

The environmental assumptions considered by Moore *et al.* do not directly correspond to those present in vehicular networks. The modified mechanism is called Stinger, and suicide notes are called stings. Stinger deviates from suicide in the following respects:

1) Stinger temporarily prohibits devices from transmitting messages, but allows them to continue receiving and forwarding messages;
2) Stinger allows multiple good nodes to be ignored by a smaller number of devices in order to exclude a single bad node;
3) Stinger permits good devices to continue accusing bad ones even after they have issued one sting.

We now describe each of these changes, and their motivation, in greater detail.

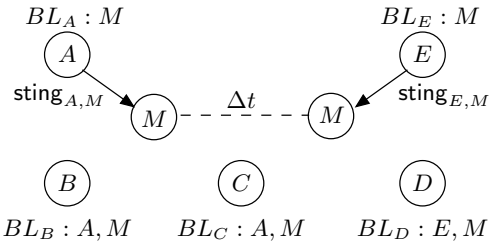

Fig. 2. Multiple Stingers for bad node B as it moves over time.

First, the original suicide mechanism proposed permanent ejection from participating on the network. Such harsh punishment is inappropriate for vehicular communications that transmit safety information. By contrast, temporary removal could be used to rapidly ignore an errant transmitter. Since most interactions are short-lived, temporary removal is equally effective in tackling misbehavior as it happens without inhibiting communication occurring much later. While the sting instruction prevents the bad and good device from sending out additional warnings, both still *receive* safety instructions from other cars. This minimizes the noticeable impact on the sacrificing driver while still penalizing a malicious device.

Second, the original suicide mechanism assumed a completely connected network. Suicide notes were broadcast throughout the network so that just one good device is removed for each bad device. Vehicular networks will be comprised of disconnected islands. High-traffic areas in cities remain separate from each other and from highways in between. Furthermore, connections are ephemeral: cars on a motorway may only be in communication range for a few seconds. Thus, it is impractical to transmit Stinger messages across a country in a short time. Instead, Stinger messages must remain localized, rebroadcast at most a few times. This keeps the response quick and minimizes communications overhead. It also means that there will be times where more than one good node has sacrificed itself for the same bad node. Yet the impact is still limited: rather than having a single node removed for one bad node, several nodes may be independently removed for one bad node. Crucially, no single device will ignore two honest nodes for the same bad node. This is because good nodes maintain a local blacklist, and they only ignore the first Stinger sender for each accused device.

Figure 2 (left) illustrates how the Stinger protocol works as cars move. Bad node M is detected by A, which broadcasts $\mathsf{sting}_{A,M}$ to instruct vehicles near A to ignore M. Hence, nodes B and C add both A and M to their local blacklists, while D and E do not because they did not receive $\mathsf{sting}_{A,M}$. As M moves into range of D and E, E issues a new removal for B, $\mathsf{sting}_{E,M}$. D adds E and M to its local blacklist, but C does not because it has already ignored M from A's sting.

This discussion motivates the third difference between suicide and Stinger: good devices continue to accuse bad ones even after they have issued one sting. This is necessary to prevent a so-called *motorway attacker* who widely broadcasts misbehavior and moves around quickly to attract many stings

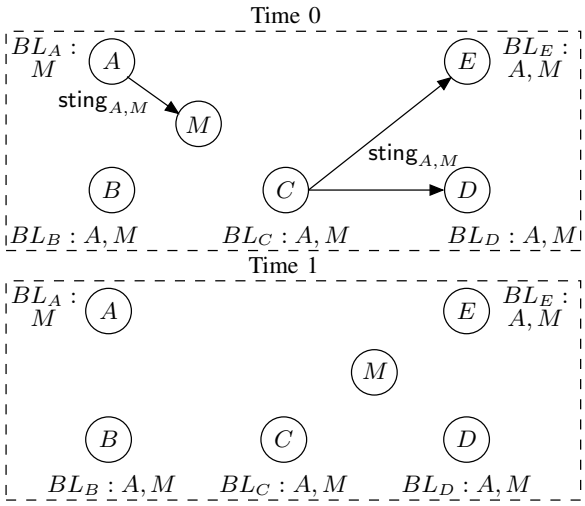

Fig. 3. Rebroadcasting A's sting against M.

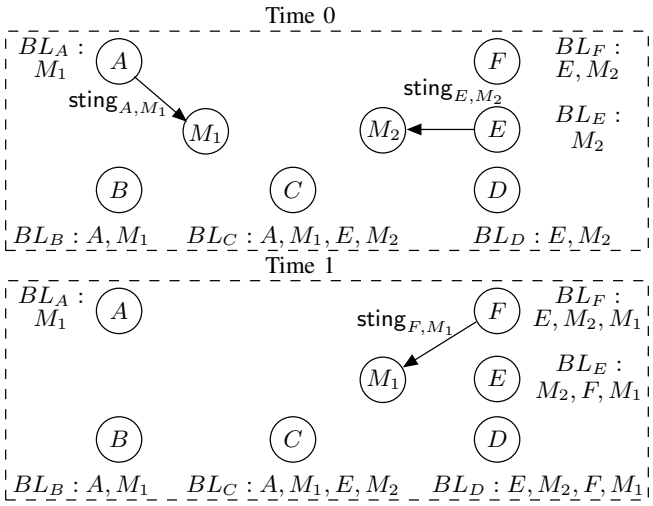

Fig. 4. Adverse impact of Stinger. Two bad nodes M_1 and M_2 are removed by A and E; E cannot warn his neighbors about M_1 at time 1 since he has already done so for M_2.

and prevent good nodes from excluding subsequent attackers.

Sting messages are locally transmitted, and they may also be rebroadcast to warn devices in case the bad device later moves in other directions. The effect of sting retransmission is shown in Figure 3. At time 0, bad node M is detected by A which transmits $\mathsf{sting}_{A,M}$. Nodes B and C then retransmit the message, notifying D and E. When M moves near to D and E at time 1, M is already ignored by them.

So what is the cost of Stinger besides message and transmission overhead? Good devices that have issued stings can no longer warn their neighbors if they detect another misbehaving device. In Figure 4, bad nodes M_1 and M_2 are present in different areas. Nodes A and E issue stings to locally remove them. However, when M_1 moves into the area previously occupied by M_2, E is powerless to warn its neighbors. E can try to remove M_1, but it has no effect since its neighbors already ignore E's messages. So F is left to issue sting_{F,M_1}. In the following analysis, we quantify the adverse impact of excluding honest devices by measuring any delays introduced when removing bad devices.

IV. Simulation framework

We evaluate the performance of LEAVE and Stinger under the stringent conditions required by vehicular networks. As both protocols rely on the ad-hoc operation of vehicles within short time delays, we simulate it using ns-2 [12] with the message access control layer parameters of IEEE 802.11p. Consequently, our simulation takes into account subtleties such as non-symmetric message reception and timing differences due to node mobility. LEAVE is implemented with an exclusion coefficient of 0.5 and a disregard threshold of 4, as in the original LEAVE paper. Each simulation run lasts 200 seconds. While stings are designed to be temporary, for the purposes of the short simulation, their effects are treated as permanent.

We now describe how we realistically model dynamic traffic conditions, attacker behavior, and the operation of detection mechanisms. We identify key characteristics which we vary in the simulations in order to better understand their impact on LEAVE and Stinger's security and performance.

A. Modeling dynamic traffic conditions

To simulate different traffic conditions, we vary:

1) average vehicle speed,
2) average vehicle density.

Presently, we use a city traffic model proposed by Saha and Johnson [17]. In Sections V-B and V-C, we simulate 150 vehicles traveling at 60 km per hr on a 2.4km by 2.4km area modeled after a city. We then vary the density and speed of cars in Section V-D. The ns-2 framework used for developing the simulations is available at [18].

B. Modeling errant behavior and its detection

In the original LEAVE paper [16], attacker behavior and detection is modeled in a simple fashion. Only one bad node participates in the system during simulations. Any device within transmission range of the bad node is deemed vulnerable to attack. Furthermore, good devices can detect bad ones as soon as they are within transmission range. We improve the simulation framework by generalizing the attacker model to allow for a more capable adversary, as well as arriving at a more realistic approximation of attacker impact and detection.

We also note that as part of its system model, the original LEAVE paper assumes an honest majority. In our simulations we allow for circumstances where this is not true. Even though the majority of all cars is likely to remain honest, it is quite reasonable for adversary-controlled devices to reach temporary, localized majorities.

To vary misbehavior, we tweak the following parameters:

1) the number of attacker-controlled devices,
2) false information dissemination versus exclusion mechanism abuse,
3) attacker impact range.

We allow the adversary to simultaneously compromise a number of devices. Attacker-controlled devices can cooperate and share information. Under *false information dissemination* attacks, the adversary attempts to cause accidents or divert congestion by sending fake safety messages. Malicious nodes disregard all bad messages originating from other nodes (i.e., they do not accuse other malicious nodes). Under *exclusion mechanism abuse*, the adversary additionally tries to disrupt the transmission of safety messages using the exclusion mechanism itself (LEAVE or Stinger). Malicious nodes falsely accuse all non-malicious nodes in communication range. As the nodes move, they discover who their new neighbors are and then vote against them. In this way, honest nodes may be implicated. Bear in mind that once an honest node has detected a malicious node as such, it ignores its votes. However, there may be a window of time where an undetected bad node can trick good nodes into believing that other good nodes are bad. Exclusion mechanism abuse may not be in the interest of an attacker whose aim is to remain undetected.

One of the most difficult aspects of simulating attack is determining which devices are made vulnerable by compromised devices. We approximate vulnerability to attack by proximity to compromised devices. The closer a device is to an errant transmitter the likelier it is to be harmed. Hence, in our simulations we vary the maximum distance from a bad node where a good node is still vulnerable.

To model detection mechanisms, we vary these parameters:

1) range of revealed misbehavior,
2) false positive rate,
3) false negative rate.

These properties are general to all detection mechanisms, and we can vary each without restricting our simulations to using a particular type of detection mechanism.

The simplest approach (taken by the original LEAVE paper) assumes that good nodes can detect bad ones with 100% accuracy so long as they are within the bad node's maximum transmission range. This behavior does not correspond to how detection mechanisms might actually work. In many cases, misbehavior can only be detected much closer to an adversary (e.g., if another car's sensors can directly observe the other car or its environment). To account for this, we can restrict detection to nodes within a specified distance. Typically, the maximum range of detection will be less than the maximum range of attacker impact. For example, a car sending a fake crash warning message is likely to be detected by its immediate neighbors, but cars further away will still be affected precisely because they believe the misinformation and react to it.

Another unfortunate property of detection mechanisms is their susceptibility to error. A false positive occurs whenever the detection mechanism flags a good device as bad. Similarly, a false negative happens when the detection mechanism mistakes a bad device for a good one. In our simulations we vary the likelihood of false positives and negatives over time. For instance, a false positive rate of 10% means that there is a one in ten chance of a false accusation per minute of interacting with other devices.

V. Simulation analysis

In this section we compare the security and performance of LEAVE and Stinger while varying the parameters just discussed.

A. Security and performance metrics

We compute three metrics:

1) average time good devices are vulnerable,
2) average percentage of good neighbors that are ignored,
3) average number of messages received per device.

The first two metrics describe the security properties of the protocols, while the third is used to compute overhead.

Most envisioned attacks on vehicular networks are time-critical – they spread misinformation that causes a car to quickly make an incorrect decision. Hence, bad devices must be detected and removed very quickly. We measured the average time good devices are vulnerable to attack by a bad node. Here, vulnerability is defined as being within transmission range of an unblocked bad device.

The second security metric we measure is the average percentage of good neighbors that are ignored due to the exclusion protocol. Each ignored neighbor reduces the number of devices that can transmit safety information, as well as participate in the Stinger or LEAVE exclusion mechanisms. This metric is usually more important for Stinger, since good devices forgo participation in order to remove bad devices. However, it is still possible for good nodes to be excluded in LEAVE (due to accidental or malicious voting against good devices).

The final metric offers a useful comparison of the work required for each strategy, since each message requires a cryptographic verification operation.

B. Detection mechanism parameters

The detection mechanism's capabilities can directly impact the effectiveness of the exclusion mechanism. To demonstrate this, we computed the security metrics while varying the detection mechanism's characteristics under the assumption that only one device is compromised (see Figure 5).

Figure 5 (top left) shows how the time devices are vulnerable varies depending on the maximum distance for which bad devices can be detected. Devices can transmit up to 300 meters. Ideally, we would like the detection mechanism to trigger as soon as a good device comes within this transmission range. In this case, devices are vulnerable for 2.9 seconds before ignoring bad devices using LEAVE. Unsurprisingly, Stinger is faster (0.6 seconds with no rebroadcasts).

As we reduce the maximum range for which misbehavior can be detected, the time exposed increases. For LEAVE, the vulnerable time increases fastest, to 11.4 seconds for 200m and 25.8 seconds for 100m. For Stinger without retransmission, the lag is 4.1 seconds for 200m and 8.6 seconds for 100m. Increasing the number of rebroadcasts further minimizes the

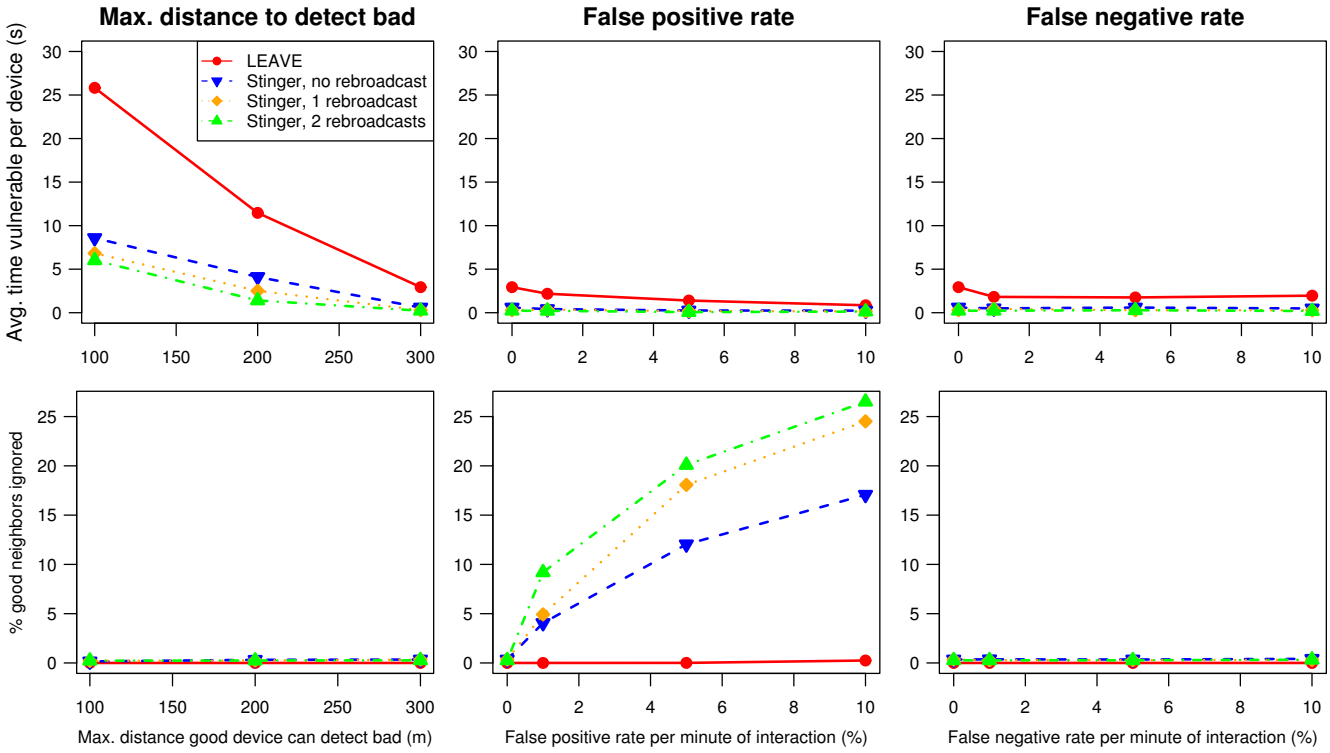

Fig. 5. Security and performance costs introduced by imperfections in the detection mechanism.

time exposed at the expense of additional message overhead. Both LEAVE and Stinger do shorten the overall exposure time, which is 29.8 seconds. Notably, LEAVE barely helps when the detection is not very good (25.3 seconds exposure for 100m maximum detection range).

Varying the maximum detection range has no impact on the proportion of good neighbors ignored (Figure 5 (bottom left)). This is not surprising, given that reducing the detection range only enables more bad devices to go undetected, which does not trigger false accusations against good nodes. For Stinger, just 0.3% of a node's good neighbors are ignored on average. This proportion is so small because the devices issuing stings are likely to change neighbors frequently. Increasing the false positive rate, by contrast, does cause good devices to ignore each other more often. Figure 5 (bottom center) shows that the percentage of good neighbors ignored increases significantly as false positives are more likely. Notably, false positive rates of up to 10% do not cause problems for LEAVE, so long as false positives are not correlated.

Recall from Figure 5 (top left) that the time vulnerable to attack decreases as stings are rebroadcast. In Figure 5 (bottom center), the ordering is reversed. Rebroadcasting stings causes more good devices to be ignored. With a 5% chance of false positive per minute of interaction, 12% of a device's good neighbors are ignored when stings are not retransmitted, 18% are ignored with one retransmission and 20% ignored with two retransmissions. Hence, there is a direct trade-off between speed of excluding bad devices and the number of good neighbors ignored.

The most noteworthy observation from the two graphs in Figure 5 (right) is that increasing the false negative rate to 10% has almost no impact. While not shown in the graph, the same holds as the share of attacker-controlled devices increases to 10%. While the time vulnerable slightly decreases for LEAVE in the top center and top right graphs, the change might be attributed to variation between simulation runs.

C. Adversary strategies

We now vary adversarial capabilities and strategies (Figure 6). The left-hand side graphs measure the effects of false information dissemination (i.e., a strategy that does not attempt to abuse the exclusion mechanism), while the right-hand side measures exclusion mechanism abuse where the goal is to cause as much damage using the exclusion mechanism as possible. In both cases, we set the maximum distance for detecting bad devices to 200m with no chance of false positives or negatives.

We simulated situations where the adversary controls from 1 device up to 15, which is 10% of all devices. This does not mean that an attacker has compromised 10% of an entire country's vehicles, which is unrealistic for all but the strongest adversaries. Rather, compromising 10% of the vehicles in a localized region (in our simulations, a 5.8 km^2 area) is quite reasonable for relatively capable adversaries.

As the proportion of attacker-controlled devices increases, the time each good device is vulnerable increases (Figure 6

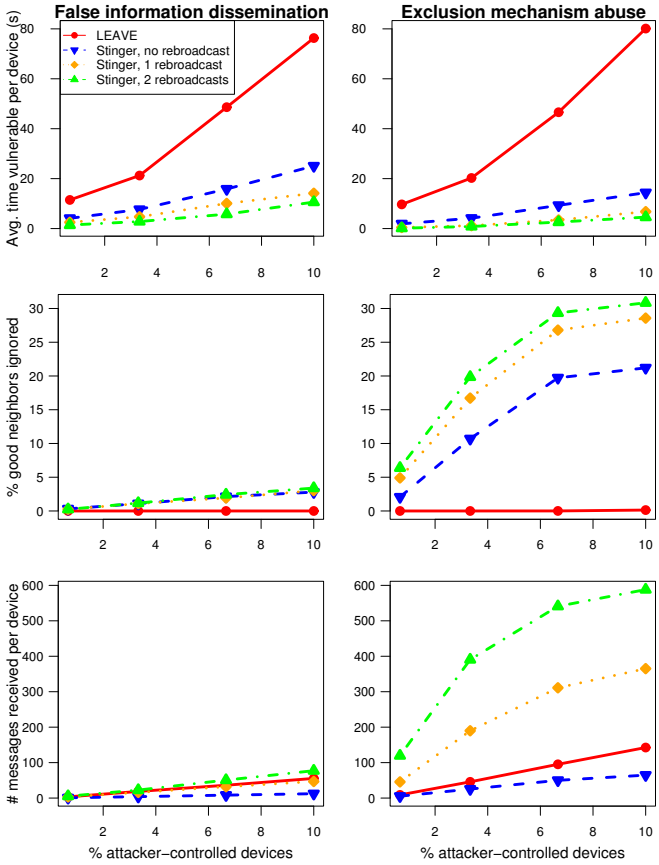

Fig. 6. Security and performance costs as the attacker controls more devices. The left-hand side graphs measure the false information dissemination strategy, while the right-hand side graphs measure exclusion mechanism abuse.

(top)). Consistent with Figure 5 (top left), LEAVE keeps devices vulnerable for longer than Stinger does. Note from Figure 6 (top right) that the vulnerable time actually decreases when the attacker is actively abusing Stinger. This is because maliciously transmitting stings instructs good devices to ignore the bad one. When the attacker's aim is to remain undetected, false information dissemination is preferred.

By contrast, Figure 6 (middle) and (bottom) explain why an attacker might prefer abusing the exclusion mechanism. When using a false information dissemination strategy, the proportion of good neighbors ignored remains small: 2.8% of good neighbors are ignored using Stinger without rebroadcasts when 10% of devices are controlled by the adversary (Figure 6 (middle left)). Under an exclusion mechanism abuse strategy, the proportion of ignored good neighbors jumps to 21.2% (Figure 6 (middle right)). This proportion is still much smaller than the worst-case scenario. With 10% attacker-controlled devices, there could be more bad devices present than a good device has neighbors. Hence, we might expect all devices to be ignored, but the simulations do not bear this out.

Why not? In the worst case for collateral damage caused by active Stinger abuse, an attacker continually broadcasting stings against every neighbor can trick every good device to ignore one honest neighbor. With luck, this one ignored honest neighbor could remain in communication range permanently. This is hard to achieve in practice since cars move and naturally change neighbors. Furthermore, it is very difficult for an attacker to locally interact with every other car to determine a legitimate neighbor to remove. Therefore, it is not surprising that the proportion of ignored good neighbors plateaus as the attacker controls more devices, as indicated in Figure 6 (middle right).

As mentioned in Section III-A, the worst-case scenario for abusing LEAVE is a roaming pack of attackers. So long as the number of attackers exceed the voting threshold, they can quickly eject any device at will. We did not simulate this situation because the outcome is clear: all good devices are removed. Instead, we simulate the situation where attacker-controlled devices vote against their neighbors continuously, but in an uncoordinated fashion. As can be seen from the simulations, uncoordinated malicious voting is completely ineffective.

Figure 6 (bottom) compares the number of messages received when using LEAVE and Stinger for a range of rebroadcasts. So long as the adversary is not attempting to abuse the eviction mechanisms, Stinger requires fewer messages than LEAVE whenever stings are only rebroadcast one time, or not at all (Figure 6 (bottom left)). Beyond that, LEAVE is more efficient in terms of message overhead.

When abusing the exclusion mechanism, however, the impact on overhead changes dramatically (Figure 6 (bottom right)). Stinger without rebroadcasting remains the most efficient strategy, but adding rebroadcasting leads to a huge increase in overhead. With 10% of the devices under adversary control, Stinger without rebroadcasting requires each device to receive 65 messages, compared to 311 with a single retransmission and 588 with two retransmissions. This dramatic difference provides further evidence that using Stinger without retransmissions is the best approach.

D. Traffic conditions

The analysis so far has considered a single, typical traffic scenario. But what happens when traffic conditions change? We varied both the density of traffic and the average speed of vehicles.

Figure 7 plots the results with 10 bad devices present and a 200m maximum distance for detecting bad devices. Increasing the density of traffic has a negative effect on LEAVE and a slightly positive effect on Stinger. Surprisingly, the average time devices are vulnerable increases as more cars are present when using LEAVE (Figure 7 (top left)). Under Stinger, by contrast, the time vulnerable decreases slightly as the density increases.

The number of messages that must be processed also increases significantly under LEAVE, while remaining steady when using Stinger (Figure 7 (bottom left)). This is because the number of other cars within communication range of each vehicle increases under higher densities. More neighboring vehicles means more broadcast warning messages from each

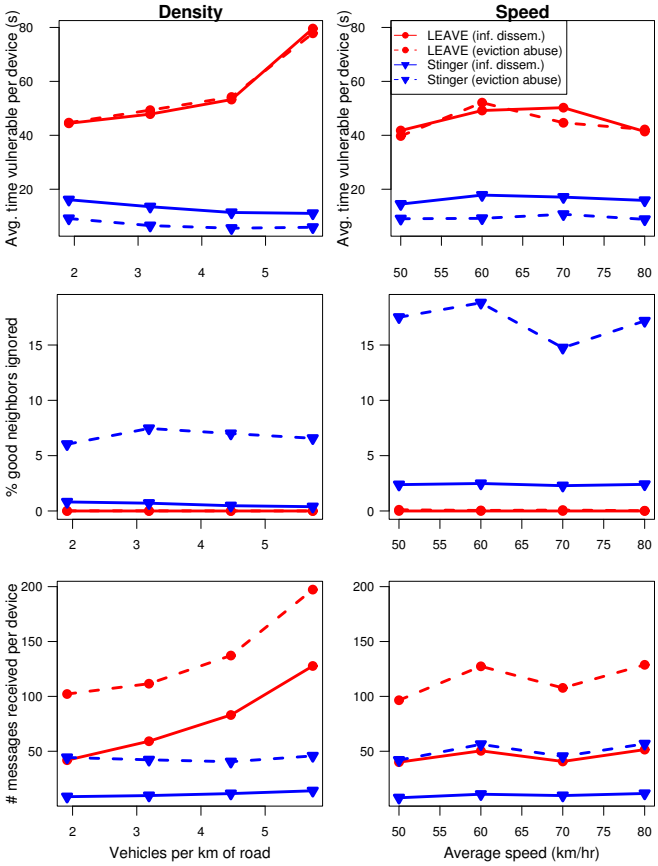

Fig. 7. Security and performance costs while varying traffic conditions.

nearby device. Since only one vehicle (or a few in the event of a collision) issues a sting for bad devices, there is no increase in overhead with increased traffic.

The right-hand side graphs in Figure 7 measure the effects of increasing the speed of vehicles, and therefore, reducing the average connection time between vehicles. From these graphs, it appears that speed does not have a significant impact on the security or performance of Stinger and LEAVE.

VI. Hybrid, adaptive exclusion strategies

The analysis in this section has identified a number of trade-offs between LEAVE and Stinger for different scenarios. To summarize:

- LEAVE is more resilient to non-zero false positive rates.
- Fewer good nodes are ignored using LEAVE than Stinger when an attacker abuses the exclusion mechanism.
- Stinger excludes bad devices faster, leaving a shorter vulnerability window.
- Stinger scales better as the density of vehicles increases.

Unfortunately, there is no clear winner between LEAVE and Stinger. LEAVE does a better job of handling false positives than Stinger, while Stinger is significantly faster than LEAVE when removing bad devices under every condition we tested. Speed of removal is critical to limit the transmission of misinformation.

We conclude that a hybrid strategy that adapts to system parameters and the environment would be able to get the best of both mechanisms, if applied properly. More precisely, three main factors affect the choice of the exclusion mechanism: tolerable vulnerability window, false positive rate, and the detected percentage of attackers. These factors are in turn influenced by the safety application in question (e.g., the vulnerability window of a hazard-warning application depends on the distance to the accident), the misbehavior being detected (e.g., mild tampering with location information is not easily detected) and traffic density. Hence, we propose the following hybrid strategy that takes these three factors into account:

1) While the vulnerability window achieved by LEAVE is tolerable, use LEAVE. Since detection mechanisms are unlikely to have zero false positive rates, Stinger will ignore a certain percentage of good neighbors.
2) When the tolerable vulnerability window is bypassed (e.g., due to increasing traffic conditions in the car's local view), switch to Stinger, but still allow devices to vote. If enough votes come in, then the stinging node can be allowed to participate again. This may help to reduce the number of ignored neighboring devices (see Figure 5 (bottom center) and Figure 6 (middle right)).

The hybrid strategy can be expressed by the following algorithm:

```
1: while tol_vuln_window ≥ vuln_window_LEAVE do
2:     run LEAVE
3: end while
4: if tol_vuln_window < vuln_window_LEAVE then
5:     repeat
6:         run Stinger
7:         blacklist stinging node
8:     until threshold_LEAVE is reached
9:     if attacker excluded by LEAVE then
10:        release stinging node
11:    end if
12: end if
```

Vehicles need not synchronize their choice of exclusion algorithm; rather, devices locally decide whether to run Stinger or LEAVE, while honoring the actions of others. Suppose device A is running LEAVE and it observes sting_{B,M_1}. A ignores both B and M_1, but A still sends out warning messages using LEAVE if it observes another malicious device M_2. If enough other vehicles using LEAVE warn M_2, then $\mathsf{disregard}_{M_2}$ messages are broadcast and recognized by vehicles using Stinger.

The evaluation of this algorithm requires testing several applications in variable traffic scenarios. We leave this endeavor to future work. A particular challenge to be explored further is locally estimating vulnerability windows.

VII. Related work

The literature on vehicular networks already contains methods for detecting bad devices. For example, Leinmüller *et al.* propose threshold-based tests to verify positioning information in vehicular networks [10]. In [6], a general framework for detecting malicious data detection compared received data to a vehicular network model.

Techniques for removing bad devices from a network often fall under the broad category of *revocation*. Revocation has been considered mostly in the context of the wired Internet and the design of Public Key Infrastructure (PKI) services [7]. Disseminating revocation information across vehicular networks, given their size and volatility, would be impractical using the same methods proposed for the fixed infrastructure.

Most existing works on vehicular network security [14], [15], [20] have proposed 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. In the context of vehicular networks, the IEEE 1609.2 Draft Standard [8] does refer to certificate revocation. It has proposed the distribution of CRLs and short-lived certificates, but does not elaborate how to achieve this. The paper proposing LEAVE [16] also described a more comprehensive revocation strategy that leverages the infrastructure to efficiently distribute revocation lists from CAs. In this context, LEAVE is a fast, temporary exclusion mechanism which triggers a slower, permanent revocation by the CA. Stinger can also be used for this purpose.

In this paper, we have distributed the task of temporarily excluding bad devices to untrusted vehicles to improve timeliness, while keeping the CA's substantial responsibilities centralized. Others have distributed the CA's responsibilities in different contexts. Zhou and Haas investigated CAs for use in mobile ad-hoc networks, distributing their functionality across a number of servers [21]. However, this scheme does not consider the problem of revocation, especially in a highly mobile environment like a vehicular network. Splitting up CA responsibilities over impromptu coalitions of devices (e.g., [5], [9]) is similar in motivation to the voting structure of LEAVE. Chan *et al.* [3] distribute static votes for excluding bad devices from a wireless sensor network. Unfortunately, threshold cryptography is of limited use whenever the voting coalitions are as dynamic as the short-lived neighbors in a vehicular network.

VIII. Conclusions

In this paper, we compared two protocols, LEAVE and Stinger, for excluding misbehaving or faulty devices from ad-hoc networks. We applied them in the context of vehicular networks where fast exclusion is both critical and hard to achieve, given the ephemeral properties of the environment. Based on a detailed simulation analysis, we found that both protocols have unique advantages and disadvantages. In particular, Stinger is faster than LEAVE and scales better with increasing traffic density, but LEAVE is more resilient to false positives and higher percentages of attackers abusing the exclusion mechanism. We therefore devised a hybrid algorithm, allowing devices to choose the strategy that best suits the circumstances. We leave the evaluation of this algorithm to future work. We are optimistic that a mixed strategy for excluding bad devices might perform better than a single single strategy in the highly dynamic environment characteristic of vehicular networks.

Acknowledgments

We would like to thank Richard Clayton, Dan Cvrček and Ignacio Llatser for their helpful comments. Tyler Moore is supported by the UK Marshall Aid Commemoration Commission and by US National Science Foundation grant DGE-0636782.

References

[1] Advanced Safety Vehicle Program, http://www.ahsra.or.jp/demo2000/eng/demo_e/ahs_e7/iguchi/iguchi.html

[2] ASTM E2213-03, 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.
tma

[3] H. Chan, V. D. Gligor, A. Perrig, and G. Muralidharan, "On the distribution and revocation of cryptographic keys in sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, pp. 233–247, 2005.

[4] Communications for eSafety, http://www.comesafety.org/

[5] C. Crépeau and C. Davis, "A certificate revocation scheme for wireless ad hoc networks," in *1st ACM Workshop on Security of Ad-hoc and Sensor Networks (SASN)*, 2003, pp. 54–61.

[6] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *1st ACM International Workshop on Vehicular Ad hoc Networks (VANET)*, 2004, pp. 29–37.

[7] R. Housley, W. Polk, W. Ford, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," Internet Engineering Task Force RFC 3280, 2002.

[8] IEEE P1609.2 Version 1, Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages (in development).

[9] 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.

[10] T. Leinmüller, C. Maihöfer, E. Schoch, and F. Kargl, "Improved security in geographic ad hoc routing through autonomous position verification," in *3rd ACM VANET*, 2006, pp. 57–66.

[11] T. Moore, J. Clulow, S. Nagaraja, and R. Anderson, "New strategies for revocation in ad-hoc networks," in *4th European Workshop on Security and Privacy in Ad-hoc and Sensor Networks (ESAS)*, Springer Lecture Notes in Computer Science (LNCS), vol. 4572, pp. 232–246, 2007.

[12] Network Simulator, http://www.isi.edu/nsnam/ns/.

[13] 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.

[14] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *ACM Workshop on Hot Topics in Networking (HotNets)*, 2005.

[15] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *3rd ACM SASN*, 2005, pp. 11–21.

[16] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE Journal on Selected Areas in Communication*, vol. 25, no. 8, pp. 1557–68, 2007.

[17] A. Saha and D. Johnson, "Modeling mobility for vehicular ad-hoc networks," in *1st ACM VANET*, 2004, pp. 91–92.

[18] Security of Vehicular Networks@EPFL, http://ivc.epfl.ch

[19] Vehicle Safety Communications Project, http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/CAMP3/pages/VSCC.htm

[20] M. E. Zarki, S. Mehrotra, G. Tsudik, and N. Venkatasubramanian, "Security issues in a future vehicular network," in *European Wireless*, 2002.

[21] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.