Non-Interactive Detection of Malicious Vehicular Network Data

G. Di Crescenzo, Y. Ling, T. Zhang and S. Pietrowicz

Telcordia Technologies, Piscataway, NJ, USA;
e-mail: {giovanni,lingy,tao}@research.telcordia.com, spietrow@telcordia.com

Abstract

Vehicular networks might be deployed in the near future, and as a consequence a potentially large number of exciting applications are expected to enhance the human driving experience. Unless the security of such applications is guaranteed, however, such enhancements may be accompanied by similarly powerful and yet undesired consequences in malicious behaviour. While current research in the vehicular networks security area has recognized conventional security and cryptographic threats, detailed modeling and analysis of threats that are specific to vehicle traffic are rarely considered in the literature. In this paper we lay ground for a comprehensive investigation of "traffic-related" threats to vehicular networks. We study the problem of modeling traffic-related attacks in these networks and present automatic and efficient (i.e., no human intervention and no expensive cryptographic protocols) yet general solutions to prevent or tolerate a number of these attacks. Specifically, we propose techniques based on the capability of implementing simple and non-interactive voting algorithms that use the mere participations of vehicles to the network and, while doing that, attempt to maximize use of already exchanged and relevant network data. We validate our techniques by providing analysis results based on both simulated and real-life mobility

Journal of Cyber Security and Mobility, 117–141.

data in typical urban environments. Previous work required interactive protocols to implement voting or consensus techniques, and implicitly left open the question we solve in this paper.

Keywords: vehicular networks, security, malicious data detection, voting protocols.

1 Introduction

The likely entrance of vehicular networks in everyday lives may encourage the development of applications in the areas of traffic safety, as well as location-specific, traffic-related, and mobility-aware services (see, e.g., [17] and references therein). For these and other applications, providing security guarantees to their communication parties requires solution to a number of challenging theoretical and practical research problems, as documented by the recent surge of publications (see, e.g., references in [3]), challenge papers (see, e.g., [10]), and projects (see, e.g., [1, 2]) in the area. In particular, a number of threats to vehicular networks are expected to surface, originating from many types of attackers with many types of goals. Current research in the vehicular networks security area [3] seems to suggest that a large number of such threats are prevented or tolerated with the use of public key infrastructures with added special properties (most notably: vehicle privacy, malicious behaviour prevention and detection). While current research has recognized conventional security and cryptographic threats to a vehicular network, these efforts have so far fallen short of investigating the types of concrete threats related to vehicle traffic, that are expected from attackers in a vehicular network. Notable exceptions are [4, 5, 8, 9, 10, 12, 13, 14], where some cases of traffic-related threats are mentioned.

Our contribution. In this paper we begin a comprehensive investigation of traffic-related threats and methods to detect, prevent and/or tolerate them. Specifically, we consider attackers trying to use vehicle communications and its interaction with various supporting sensors systems (e.g., time, position, and speed sensors, as well as the certificate management cryptosystem) to cause harm to other vehicles or the vehicular network. For example, attackers could use vehicle sensors and software clients to raise false alerts and cause harm to the vehicular network operations. We discuss the likelihood of such attacks, their impact, and how to deal with them. In particular, we investigate the problem of modeling traffic-related attacks in vehicular networks and presenting *automatic* (i.e., not relying on human intervention), *time-efficient* (i.e., not requiring additional expensive cryptographic operations) and *non-interactive* (i.e., not requiring additional message exchanges between vehicles) solutions to detect, prevent or tolerate such attacks. Our techniques are based on the capability of implementing elementary non-interactive voting methods *directly* from the traffic statistics sent by vehicle software clients during their mere participations to the vehicular network. Specifically, feedback about the validity of a vehicle's traffic data is derived from other vehicles' traffic data and combined via a non-interactive voting algorithm that allows to determine whether its data was trustworthy or not. The question of whether non-interactive voting algorithms could be used, which we solve in this paper, was implicitly left open by research in recent works, such as [3, 11, 13], that defined and analyzed interactive voting or consensus protocols to deal with specific classes of traffic-related threats. Additionally, in previous work, interactive voting protocols were executed among neighbor vehicles, which were not necessarily relevant to the traffic scenario at hand. In our protocols, we restrict the set of protocol participants using a simple greedy algorithm that optimize participant relevance and usage of traffic data from them. We stress that in this context, given the challenging solution constraints due to vehicle mobility and real-time effectiveness, both algorithm efficiency and simplicity are very desirable properties. We first present general voting algorithms that are applicable to a very large class of abuse cases, and then detail these algorithms to specific abuse cases, such as false speeding, false congestion, false braking, false timing and position data, and higher message frequency. We present our analysis results for the seemingly most challenging abuse case (i.e., false braking) for which, using simulated traffic mobility data on a real geographic map as well as real-life traffic mobility data, we see that the voting algorithms have high percentage of detecting abuses.

Organization of the paper. In Section 2 we present our modeling of traffic-related threats, malicious behavior and a number of abuse cases of specific interest. In Section 3 we present our general techniques to detect malicious data, including our voting algorithm and our simple solution to the problem of selecting the most relevant set of neighbor vehicles whose traffic data can be used in each vehicle's voting algorithm. In Section 4 we detail these general techniques for a number of concrete abuse cases of specific interest. In Section 5 we present our analysis results for one specific abuse case, using simulated traffic mobility data on a real geographic map as well as real-life traffic mobility data.

History of results. This paper is the journal version of our preliminary conference publication [18], enhanced in a number of ways: we have significantly improved our voting algorithm by considering the new problem of selecting the set of neighbor vehicles so to optimize data relevance for a vehicle's voting algorithm; moreover, we have expanded our analysis results, previously only based on simulated traffic mobility data on a real geographic map, by additionally using real-life vehicle mobility data.

2 Model, Malicious Behaviour and Abuse Cases

2.1 Vehicular Network Model

We consider the following entities: a large number of vehicles with On Board Equipment (OBE) including time, position and speed sensors, and cryptographic software and hardware; a limited number of Road Side Equipments (RSE) that might act as servers and are situated in carefully-chosen geographic locations so to be able to manage all vehicles in their pre-assigned geographic area, using wireless communication. RSEs may, or, most likely, may not cover the entire geographic map. Naturally, we also consider a backbone network that communicates with RSEs and, in special situations with OBEs. We assume that a public-key infrastructure is in place, and that the vehicle-to-vehicle communication is authenticated using digital signatures and certificates managed by a back-end certificate authority (CA), possibly using RSE as front-end CA servers (see, e.g., [6] for more details on how such an infrastructure could be deployed).

2.2 Traffic-Related Malicious Behaviour

Malicious OBE behaviors related to traffic activity include attempting to use the compromised vehicle clients to send malicious messages to other vehicles or RSEs; incapacitate crucial vehicular network functions (e.g., its key and certificate management system and its privacy preserving functions); harm, disrupt, or destroy portions of the vehicular network, and public images of network providers and vehicle manufacturers.

Scenarios for OBE Malicious Behavior. The following list focuses on specific malicious behavior from OBEs based on communication within the vehicular network's activities: (1) Sending probe/heartbeat messages that are inaccurate or carefully modified so that: false announcements of traffic conditions in a given area cause a redirection of traffic that benefits the attacker's business; incorrect driving conditions or patterns to create automobile accidents, possibly motivated by potential insurance fraud or just sadistic pleasure; government agency decisions on road improvements or traffic management are affected; the attacker is given preferential treatment by the vehicular network for the purposes of evading law enforcement, assisting in criminal operations, or diverting attention from a primary attack; a new virus is launched or a new vulnerability in the vehicular network is found. (2) Modifying the functioning of a vehicle's OBE to carry out attacks, such as one of the above ones, or because of privacy concerns; specifically: modifying timing intervals of multiple probe or heartbeat messages; delaying the delivery of probe or heartbeat messages; sending more probe or heartbeat messages than the OBE is supposed to; not sending probe or heartbeat messages for a long enough time interval; disabling the functioning of a vehicle's OBE because of privacy concerns.

2.3 Concrete Abuse Cases

Naturally, it is of interest to consider some of the most important scenarios in the above list and detail them into individual abuse cases, for which we can carry out a formal and rigorous investigation of the problems of detection and prevention of malicious behavior. As main examples in the rest of the paper, we will use abuse cases based on malicious modifications of heartbeat data, such as false speeding (e.g., vehicles pretending to be speeding on their lane), false congestion (e.g., vehicles pretending that a certain geographic area is congested), false braking (e.g., vehicles pretending to be abruptly braking), false timing and position data (e.g., vehicles sending incorrect time and position data in their heartbeat messages), and higher message frequency (e.g., vehicle sending a higher number of heartbeat messages than they are supposed to).

3 General Detection Techniques

We present formal modeling and formally defined techniques to efficiently detect these malicious behaviors in all abuse cases mentioned in the previous section. To better meet the communication challenges expected in vehicular networks (i.e., ad-hoc connectivity, fast client mobility, partial or lack of server coverage, etc.), our technique are specifically based on time-efficient and conceptually simple tests that can be performed by an OBE, helped by vehicle sensors, and possibly an RSE, or a network server connected to a RSE, using data from a carefully selected set of neighbor vehicles.

3.1 An Informal Discussion

To clarify the difficulties that we face in coming up with such tests, consider, as an example, the problem of detecting a vehicle that, while contributing traffic-related data, triggers a false car accident alert in a given geographic position at a given time. It would be very easy for other vehicle owners in the area to take a look at the claimed geographic position, notice that actually no accident has happened, and communicate this fact to the CA that would thus revoke the attacker's keys and eventually take appropriate actions against its vehicle. Naturally, this solution is impractical, as it requires human intervention. Alternatively, one might want to let the vehicles in the area run a secure distributed agreement protocol, where the vehicles agree on whether an accident really happened or not. This idea is unlikely to result in a correct or feasible solution because of various difficulties, including the combination of limited radio range and vehicle mobility, which prevent this protocol's successful termination, and vehicle clients' computational constraints. Thus, we investigate the problem of presenting *automatic* (i.e., not relying on human intervention), *time-efficient* (i.e., not requiring additional expensive cryptographic operations) and *non-interactive* (i.e., not requiring additional message exchanges between vehicles) solutions to prevent or tolerate traffic-related attacks to vehicular networks. Our solutions consider a two-phase approach to deal with each specific abuse:

Alert Generation Procedure. In response to one or multiple probe or heartbeat messages sent by neighbor vehicles, a vehicle uses this procedure to generate an alert for its driver.

Abuse Detection Procedure. In response to a generated alert, this procedure carefully selects a set of neighbor vehicles to potentially detect whether the alert was an abuse.

3.2 Formal and Detailed Description

At any given time, each vehicle A sends a stream of probe or heartbeat messages, where each message is denoted as *pm* and contains, among other things, a certificate *cert* and a tuple (*t, sp, pos, dir*), where *t* denotes a timestamp, *sp, pos, dir* denote the vehicle's claimed current speed,position, direction, respectively. Some abuses would generate an ordered sequence of such messages.

Any message or sequence of messages may or may not generate an alert depending on whether the Alert Generation Procedure finds something worth generating an alert or not in them. If an alert is not generated, then nothing happens. If an alert is generated, an abuse may or may not be detected, depending on whether the Abuse Detection Procedure finds that the alert was false (and was thus an abuse) or not. Both the Alert Generation Procedure and the Abuse Detection Procedure may be executed by each vehicle and may use all the communication received by the vehicle from other vehicles in its radio range (or a carefully selected subset of them).

We say that any message *pm* is an *alert message* if it causes the Alert Generation Procedure to generate an alert and thus the execution of an Abuse Detection Procedure. Similarly, we can define a *sequence of alert messages*. For generality, in the rest of our discussion, we will only consider sequences of alert messages.

Alert Generation Procedure: Upon receiving a sequence of messages $pm[1], \ldots, pm[k]$, a vehicle may issue an alert to the driver or the vehicle control systems if some specific (and anomalous) instances of the following 3 types of conditions are simultaneously satisfied: a speed condition (e.g., a condition based on the values $sp[1], \ldots, sp[k]$); a geography condition (e.g., a condition based on the values $pos[1], \ldots, pos[k]$); and a time condition (e.g., a condition based on the values $t[1], \ldots, t[k]$).

If an appropriate combination of these conditions is satisfied (the details of which may depend on the specific abuse), an alert is sent to the driver or vehicle control system. This can happen in a few ways, according to how time, speed and geography conditions are instantiated and according to whether the alert is generated with or without the help of an RSE. In the rest of the paper, we consider the harder case where this happens without the help of an RSE. Specifically, every vehicle v receiving a sequence of alert messages $pm[1], \ldots, pm[k]$, issues an alert for its own driver after testing the time, speed and geography conditions. In this mode, each vehicle needs to be able to evaluate local time, speed, and geography conditions without any help from an RSE. One can think of many ways for vehicles to enter false speed, geography or time values in their messages, and thus to abuse such a mechanism into causing the generation of false alerts. The following procedure attempts to detect these situations.

Abuse Detection Procedure: This procedure is intended to evaluate whether the above Alert Generation Procedure was the result of a real alert or of an abuse and to identify the certificates used in the abuse, which can then be used to trigger the certificate management system to revoke the certificates. While each vehicle runs the Alert Generation Procedure on a single sequence of messages (possibly sent by a single vehicle), in the Abuse Detection Procedure each vehicle can use messages received from some or all of the vehicles in its radio range, and can thus use additional information to come up with an improved decision on whether the alert was real or not. The fundamental assumption we make is that in any sufficiently large vehicle neighborhood, the majority of the vehicles will return correct assessments showing that the alert generating vehicle's assessment is or is not a correct one. As with the Alert Generation Procedure, the Abuse Detection Procedure can happen in a few ways, according to how the time, speed and geography conditions are evaluated on each vehicle and to which messages are received by each vehicle. Specifically, every vehicle v that receives m sequences of messages $pmseq[1], \ldots, pmseq[s]$, such that $pmseq[j] = pm[j, 1], \ldots, pm[j, k]$, for $j = 1, \ldots, s$ (one or more of which being sequences of alert messages), reaches a decision on whether the alerts can be associated to an abuse or not. If yes, the certificates associated with the abuse are deemed misused and treated as untrustworthy thereafter. Future messages carrying an untrustworthy certificate can be discarded. When the vehicle has connectivity with the infrastructure network and CA again, it will report such certificates to the CA along with a transcript of the alert messages $pmseq[1], \ldots, pmseq[m]$, which could in turn trigger the CA to revoke the certificates. This decision is then broadcast to all vehicles in the area along with a transcript of the alert messages $pmseq[1], \ldots, pmseq[m]$. Here, the procedure uses some type of (authenticated) voting scheme where messages from some or all neighbor vehicles are used to produce a vote (one per vehicle) that all together will confirm or refute the presence of an abuse. Should a more interactive variant be needed, this vote might also be expressed manually by the vehicle driver (e.g., by clicking a yes/no button on the OBE), or sent by the voting vehicle after specific vehicle behaviour happens or as a consequence of the alert (however such level of interaction is not required in our protocols). Finally, each vehicle will use all these votes to reach a decision on whether the alerts can be associated to an abuse or not, and then use this decision for its own safety purposes.

We now describe a generic voting scheme that can be used to support *Vehicle Abuse Self-Detection*. The scheme can be presented into two variants: (1) a *non-interactive* variant, where vehicles whose data is used in the voting scheme do not need to send any additional communication message; and (2) a *one-round* variant, where voting vehicles send one additional communication message with respect to the usual heartbeat messages sent during their typical vehicular network activity.

In what follows, we detail the non-interactive variant of the voting scheme, and then in the next section we briefly describe the modifications needed to obtain the one-round variant. We start with a simplified version of the scheme, where data from all neighbor vehicles and thus all neighbor vehicles' votes are equally relevant.

Recall that in the above described Alert Generation Procedures, a vehicle v receives from (at least) one vehicle a sequence of alert messages $pm[1], \ldots, pm[k]$, which trigger the voting procedure. After some time interval T, the messages from the m vehicles in v's neighborhood might contain useful information to assess whether the alert was due to an abuse or not. Then, based on certified messages (and included time, speed and geography values) from vehicle i, the vehicle v may derive a 0/1 vote $e[i]$ on whether the alert was legitimate or not, respectively. In other words, this vote will indicate whether, from the point of view of vehicle i (which may be different from all other vehicles in the neighborhood because of radio range differences), the alert was legitimate.

Let m be the number of votes derived by vehicle v using communication by its neighbors; then v can use the following *majority voting test* to conclude that the alert is illegitimate:

$$\mathrm{e}[1] + \cdots + \mathrm{e}[\mathrm{m}] > \mathrm{c}\,\mathrm{m},$$

for some confidence parameter $c \in [0, 1]$. Possible approaches to choose c include statistical studies depending on the abuse type and environment. If the alert is concluded to be illegitimate, then certificate *cert* used in the alert message is deemed misused.

As mentioned, the above test can be applied when all neighbor vehicles' votes are equally relevant. In practice, this is not the case as vehicles in the same radio range of vehicle v may be considered very far from v with respect to many types of traffic events. These vehicles may be not so relevant to the event generating the alert and thus it may be useless (or even misleading) to use their data to generate a vote $e[i]$ included in the above test. More generally, one can have vehicles that are more or less relevant to the event generating the alert, depending on their relative geographic position with respect to the vehicle generating the alert. In our voting algorithm, we would like to use as much as possible messages from relevant vehicles in v's neighborhood to assess whether the alert was due to an abuse or not. To formalize this problem, we define the following quantities:

Relevance coefficient w: the coefficient $\mathrm{w}[i]$ is a value in [0,1] and captures how relevant is the i-th vehicle in v's neighborhood to assess whether the alert was due to an abuse or not (here, 0 denotes no relevance and 1 denotes the maximum possible relevance). Suitable values for $\mathrm{w}[i]$ can be estimated empirically, perhaps starting from an analytical hypothesis. For instance, one could set $\mathrm{w}[i]$ as a rapidly decreasing function of the distance of the i-th vehicle from the position of the vehicle whose messages generated the alert.

Reliability coefficient x: the coefficient $x[i]$ is a value in [0,1] and captures how reliable is the data received from the i-th vehicle in v's neighborhood to assess whether the alert was due to an abuse or not (here, 0 denotes no reliability and 1 denotes the maximum possible reliability). Intuitively, the reliability of the i-th vehicle is directly proportional to the number of heartbeat messages from this vehicle that can be used to generate vote $e[i]$.

We observe that while the relevance coefficients are essentially determined by traffic, distance and geographic conditions, the reliability coefficients can be chosen by the vehicle running the voting algorithm within the abuse detection procedure. Specifically, this vehicle can choose to allocate the (necessarily limited) number of heartbeat messages that it can inspect within time T so to maximize the relevance of the voting algorithm. This gives rise to the following maximization problem:

$$\mathrm{Max}\left(\sum_{i=0}^{m}\mathrm{w(i)x(i)}\right) \quad \text{s.t.}: \ \mathrm{w(i)} \in [0,1],$$

$$\mathrm{x(i)} = \frac{\mathrm{u(i)}}{\mathrm{U}}, \quad \mathrm{u(i)} \in \{1,\dots,\mathrm{U}\}, \quad \sum_{i=0}^{m}\mathrm{u(i)} < \mathrm{W},$$

where $u[i]$ is the number of heartbeat messages from the i-th vehicle that can be used to generate vote $e[i]$, U is the maximum number of such messages that can be received by v within time T, and W is the maximum number of heartbeat messages from all m neighbors of vehicle v that can be received by v within time T.

This optimization problem is the knapsack problem with fractional weights and discrete variables. Because of the structure of the discrete variables $x[i]$, the problem can be solved using a variant of the same simple greedy algorithm that solves the fractional knapsack problem. Specifically, this algorithm repeatedly chooses the max allowed value for the variable $u[j]$where j is such that $w[j]$is the max weight yet unused, until the total number W of heartbeat messages is allocated. We note the simplicity of this algorithm that is expected to be effectively run in real-time from vehicles.

As a consequence of these considerations, we modify our previous voting test into the following. Let m be the number of votes derived by vehicle v using communication by its neighbors; then v can use the following *weighted majority voting test* to conclude that the alert is illegitimate:

$$\mathrm{w[1]x[1]e[1]} + \cdots + \mathrm{w[m]x[m]e[m]} > c(\mathrm{w[1]x[1]} + \cdots + \mathrm{w[m]x[m]}),$$

for a predetermined confidence parameter $c \in [0, 1]$, where $w[i]$ and $x[i]$ are set as specified above (that is, $w[i]$ is a decreasing function of the distance of the i-th vehicle from the position of the vehicle whose messages generated the alert, and *x[1],...,x[m]* are chosen so to maximize the above knapsack problem).

One-round voting: An alternative approach to use a voting procedure to detect abuses can be obtained if we add the ingredient of interaction between vehicles in the above paradigm. Specifically, instead of each vehicle determining whether received messages are anomalous enough to raise an alert, one could design a scheme where any given vehicle can issue an alert, based on received messages, to all of its neighbors. Then, upon receiving this alert, vehicles will use their received messages to issue a vote and send it back to the alert-generating vehicle. Finally, this latter vehicle will collect the votes and run any one of the previously defined statistical tests. The recipient of these votes will be able to link the votes to the previous alert messages, to verify how it was generated and to verify its related authenticating signature. To preserve anonymity, authenticating signatures of these voting messages are conducted using the same data authentication protocol used for authenticating probe and heartbeat messages (e.g., attaching and verifying signatures from anonymous keys and certificates). One potential advantage of this scheme is in the fact that voters can add received messages to the information used in abuse determination. On the other hand, major disadvantages include the added interaction between vehicles, and the possibility of multiple votes from the same malicious vehicle (that would require a separate security mechanism).

4 Detecting Specific Abuses: Methods and Analysis

Naturally, it is of interest to consider some of the most important individual abuse cases, for which we can define formal and rigorous procedures for detection and prevention of malicious behavior, as application examples for the techniques in the previous section. In particular, we consider 5 abuse cases, and describe case-driven factors instantiating the general approach in the Section 3: time, speed and geography conditions used to generate an alert from probe messages, and the algorithm used to derive a vote from each vehicle's heartbeat messages with respect to a given alert, during the abuse detection procedure.

4.1 Abuse Case 1: False Speeding

Consider a 2-lane road where a vehicle on the right lane gets notified if a vehicle beside or behind it on the left lane is traveling at a speed that is significantly higher than expected. Such an alert would tell the driver on the right lane not to switch to the left lane until the vehicle on the left lane moves away. If an attacker vehicle in the left lane can send false “high speed” alerts or make otherwise misreport its presence, it may force vehicles on the right lane to give way to the attacker vehicle.

False Speeding Alert Generation Procedure: Recall that upon receiving a sequence of messages $pm[1], \ldots, pm[k]$ related to a vehicle A, a vehicle v may issue an alert to the driver or the vehicle control systems if some combination of a speed condition, a geography condition, and a time condition is satisfied. Also, recall that each value $pm[i]$ contains a certificate $cert[i]$ and a tuple $(t[i], sp[i], pos[i], dir[i])$, where $t[i]$ denotes a timestamp, $sp[i]$ denotes A's claimed current speed, $pos[i]$ denotes A's claimed position, and $dir[i]$ denotes A's claimed travel direction. In the abuse case of false speeding, a speed condition can be set as follows: $sp[i] > current\text{-}speed(v) + p(1)$, for $i = 1, \ldots, k$, where $p(1)$ is some safety parameter $p(1), current\text{-}speed(v)$ is vehicle v's current speed value, and k may be set to a very small constant. Then, a geography condition can be set as follows: values $pos[i]$ and $dir[i]$, for $i = 1, \ldots, k$, reveal that vehicle A is proceeding in the left lane, behind and towards vehicle v. Finally, a time condition can be set as follows: tuples $(t[i], sp[i], pos[i], dir[i])$ contain increasing values $t[1], \ldots, t[k]$ where any two consecutive values are almost equally distant. Several variants of this setting are possible, based, for instance, on local speed limit, larger time gaps between tuples due to reception problems, etc.

False Speeding Abuse Detection Procedure: In this procedure, upon receiving a speeding alert, a vehicle also receives m sequences $pmseq[1], \ldots, pmseq[m]$ of messages, where $pmseq[j] = pm[j,1],\ldots,pm[j,k]$, for $j = 1,\ldots,m$, that are related to the alert of speeding from a single (for simplicity) vehicle A. Then the vehicle has to reach a decision on whether the alert was false or not; that is, whether all these messages can be associated to an abuse or not. To reach this decision, the vehicle derives a 0/1 vote $e[i]$ from each sequence $pmseq[i]$ by evaluating trajectory consistency with the speeding, geography and time values for vehicle A. As an example, if according to the messages received, it appears that the speeding vehicle is in the same position as other vehicles, then a trajectory inconsistency is found and the alert is considered an abuse. Once votes are obtained, the detection procedure continues as in the Abuse Detection Procedure paradigm outlined in the previous section; i.e., using a statistical test on the sum of these votes to check whether the alert was legitimate, and the previous analysis of the test success applies here as well.

4.2 Abuse Case 2: False Congestion

Consider vehicles approaching a fork in the road and receiving traffic adversary messages from a backend network application (or from other vehicles). When vehicles are informed by malicious traffic messages that congestion exists 1 mile down the right side of the fork, they will likely take the left path. One or more malicious vehicles could trigger such congestion notifications to create benefit for themselves, such as causing others to take a different route that the one the attackers want to take.

We start by considering a number of vehicles $A[1],\ldots,A[u]$, that at a given time send messages $pm[1],\ldots,pm[u]$, where the i-th message contains, among other things, a certificate $cert[i]$ and a tuple $(t[i], sp[i], pos[i], dir[i])$, where $t[i]$ denotes the timestamp for the i-th alert message, $sp[i]$ denotes $A[i]$'s claimed current speed of this vehicle, $pos[i]$ denotes $A[i]$'s claimed position, and $dir[i]$ denotes $A[i]$'s claimed travel direction. Any such $u-$tuple of messages can trigger an alert generation procedure and an abuse detection procedure, which are now described in detail.

False Congestion Alert Generation Procedure: Given any messages $pm[1],\ldots,pm[u]$, an alert may be somehow generated if a speed condition is satisfied (e.g., $sp(i) < p$, for all vehicles $A[i]$, $i = 1,\ldots,u$, and for some parameter p, where $sp(i)$ denotes vehicle $A[i]$'s current speed value) and a geography condition is satisfied (e.g., values $pos[1],\ldots,pos[u]$ and $dir[1],\ldots,dir[u]$ reveal that the associated vehicles $A[1],\ldots,A[u]$ are proceeding in the same direction and in the same geographic area, and this area is accessible through only one possible road at a fork) and a time condition is satisfied (e.g., the u messages are all generated at relatively close times). Several variants of these example conditions are possible, based, for instance, on local speed limit, larger time gaps between tuples due to reception problems, etc.

False congestion Abuse Detection Procedure: This procedure detects whether the above alert generation procedure resulted in an incorrect congestion alert and thus identify and revoke the certificates that caused it. More precisely, the procedure consists of a variant of a voting scheme, where messages from other vehicles in the neighborhood are used to confirm or refute the presence of a congested area in the specific geographic area. Each one of these votes will be directly derived by a vehicle's messages, as a consequence of the generated false congestion alert. Specifically, after a sufficiently long time interval T, the vehicles $A[1], \ldots, A[u]$ might be able to send additional probe/heartbeat messages that help any other vehicle in determining whether the alert was due to a congested area or not. Using speed, position and direction information in additional probe/heartbeat messages from vehicles $A[1], \ldots, A[u]$, any other vehicle can implement a voting scheme, as follows. For $i = 1, \ldots, m$, the vehicle associates to the i-th vehicle $A[i]$ a 0/1 vote $e[i]$, meaning that this vehicle is approaching a congested area or not, implying that the congestion alert was or was not legitimate. To obtain vote $e[i]$, the vehicle can evaluate trajectory consistency of the speeding, geography and time values for vehicle $A[i]$ with those for all other vehicles in $A[i]$'s neighborhood. As an example, if according to the messages received, it appears that one or more among vehicles $A[1], \ldots, A[u]$ are in the same position as other vehicles, then a trajectory inconsistency is found and the alert is considered an abuse. Once all votes are obtained, the detection procedure continues as in the Abuse Detection Procedure paradigm outlined in the previous section; i.e., by using a statistical test on the sum of these votes to check whether the alert was legitimate or not, and the previous analysis of the success of this test applies here as well.

4.3 Abuse Case 3: False Braking

Consider a sequence of cars driving on the same lane, and consider the relatively frequent case of a vehicle in the sequence braking hard, possibly as a result of a specific road condition, but at the same time creating a dangerous situation to vehicles traveling behind. Naturally, most cars in this sequence may have a limited visibility, of up to a few cars ahead of them, and it would be desirable that the vehicular network has in place a mechanism that notifies the cars in the subsequence behind the braking car with an associated alert message. Now, this notification would effectively suggest these vehicles to pay more attention, slow down and perhaps even brake, potentially generating smaller-impact danger situations. Thus, an attacker may be interested in claiming the presence of an abruptly braking vehicle so to generate danger or attempt to create an opening for a lane switch, even though the attacker may not be traveling in this area or not braking at all.

The basic idea for detecting a false braking attack is that, after a braking alert message is generated, some feedback is obtained from vehicles that are approaching the area of interest (just before the breaking vehicle) at about the same time that the braking probe/heartbeat message generating the alert was received. Again, this feedback is used as a vote in some kind of a voting scheme, where other vehicles only confirm or refute the presence of a braking vehicle. Interestingly, we observe that this feedback can be directly obtained by the already existing vehicular network communication. For instance, such a vote might be directly derived from the messages sent by vehicles directly following the braking (or supposedly so) vehicle.

We start by considering a vehicle A, that at any given time sends valid messages $pm[1],\ldots,pm[u]$, where the i-th alert message contains, among other things, a certificate $cert[i]$ and a tuple $(t[i], sp[i], pos[i], dir[i])$, where $t[i]$ denotes the timestamp for the i-th message, $sp[i]$ denotes A's claimed current speed of this vehicle, $pos[i]$ denotes $A[i]$'s claimed position, and $dir[i]$ denotes $A[i]$'s claimed travel direction. Any such u-tuple of messages can trigger a false braking alert generation procedure and a false braking abuse detection procedure, which are now described in some detail.

False Breaking Alert Generation Procedure: Given any messages $pm[1],\ldots,pm[u]$, an alert may be somehow triggered if a speed condition is satisfied (e.g., $sp(i) < sp[i-1]$, and $sp[i] < p$ for $i = 1,\ldots,u$, and for some parameter p, where $sp(i)$ denotes vehicle A's speed value in the i-th message sent while abruptly braking) and a geography condition is satisfied (e.g., values $pos[1],\ldots,pos[u]$ and $dir[1],\ldots,dir[u]$ reveal that the vehicles to be alerted are proceeding towards the braking vehicle A; here, this condition can be strengthened by evaluating the number of vehicles with larger speed in a sufficiently small geographic area of interest with respect to the area where the abrupt braking occurred; if this number is not 0, then this is an indication that the braking did not occur, for instance because the vehicle claimed to be braking is actually in a different geographic area). As for the previous abuse cases, several variants of these example conditions are possible.

False Braking Abuse Detection Procedure: We only sketch this procedure as it is again based on a variant of a voting scheme. Specifically, this procedure is intended to detect whether the above alert procedure resulted in an incorrect braking alert and thus identify and revoke the certificates that caused it.

More precisely, the procedure consists of a variant of a voting scheme, where messages from other vehicles are used to confirm or refute the presence of an abruptly braking vehicle. A more specific scenario is considered in greater detail in the simulation results of next section. As for the previous abuse cases, all obtained votes are used and proved to be effective as in the Abuse Detection Procedure paradigm outlined in the previous section.

4.4 Abuse Case 4: False Timing and Position Data

More generally, at any given time and on any kind of street, a vehicle may obtain different types of notifications from the vehicular network infrastructure triggered by other vehicles' probe or heartbeat messages. These notifications may cause these vehicles to take specific actions, such as taking a different route. Here, an attacker may be interested in generating false notifications even just for the sake of creating a nuisance to other vehicles or, more generally, because it increases the success probability of concrete traffic-related attacks. While in the previous abuse cases, we focused on attackers sending incorrect speed information in their probe or heartbeat messages, it remains of interest to consider the case of misrepresented values for other data, such as timestamps, position, travel direction, etc. For instance, an attacker indirectly involved in an accident, and sharing part of the guilt for it, would be very interested in sending probe or heartbeat messages with incorrect position data, signaling that the attacker's vehicle was in a different area at the time when the accident occurred. We only give an informal treatment here as a more formal treatment can be derived as done in the previous sections.

False timing information is actually easy to detect from a vehicle that puts sufficient trust in its own timing information. False position information is also easy to detect in one particular case, where the malicious vehicle claims to be in an area geographically outside of the radio range of a detecting vehicle. The following case is somewhat harder: a malicious vehicle traveling in position *pos* within the radio range of the detecting vehicle, claims to be traveling in position *pos'* that is still within radio range of the detecting vehicle. One hopes here that at the same time another vehicle is traveling in position *pos'*, thus resulting in a trajectory inconsistency. The detection of false direction information can be analyzed similarly as done for the detection of false position information as long as the detecting vehicle has the capability of tracking a given malicious vehicle (possibly without violating its anonymity) for a limited amount of time and space.

4.5 Abuse Case 5: Higher Message Frequency

Typical "traffic-safety" vehicular network activity requires vehicles to send probe or heartbeat messages with a pre-specified time frequency. An attacker could increase the frequency of its messages, especially if some of these messages contain false information, as specified in the previous abuse cases. For instance, in abuse cases 2 and 3, an attacker being able to send messages at higher frequency will be much more effective in reaching her goal. Another scenario where this may be of interest to an attacker is in performing a "denial of service attack" both against another vehicle or an RSE. Here, the attacker's goal is to prevent another vehicle from performing basic functions, such as functions related to certificate management, which may impact the vehicle's security and anonymity properties. Our techniques to detect these attacks vary depending on specific properties of the certificate management scheme used and are omitted here.

5 Detecting Specific Abuses: Analysis Results

To validate our voting algorithm in Section III, we have analyzed its effectiveness (i.e., whether some vehicles can reliably detect abuse data) on the abuse cases in Section IV using both simulations and real data. We have focused our analysis results on the false braking abuse case, but an analogue analysis for the two main other cases of false speeding and false traffic congestion did provide strictly better results. We have considered the following basic scenario for the false braking abuse case: a sequence of vehicles, denoted as A, B, C, D, and E, proceeds along the same lane in a 2-way road (see Figure 1).

At some time, vehicle A shows in its heartbeat messages some traffic-related data showing evidence that it is braking hard (and eventually stopping), thus creating a potential danger situation for all 4 vehicles right behind. If this braking claim is true, vehicles B, C, D, and E will eventually stop as well. On the other hand, if the claim is false, it would be desirable that as many as possible vehicles in the sequence quickly (i.e., before an accident is created) realize that A's claim was a false alert, using our voting technique. In this

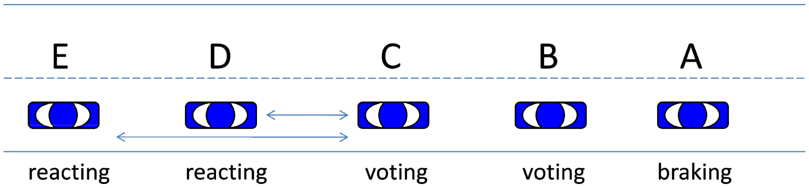

Figure 1 In a 2-way road, A's heartbeat data suggests A is suddenly braking and, using heartbeat data from B and C, vehicles D and E need to have enough reaction time to judge whether the braking claim is true or not.

example, D and E can non-interactively run our voting algorithm based on the heartbeat data received from A, B and C. Here, note that (a) if the braking claim is true, then B and C will also necessarily brake and eventually stop by the time they get to A's position, while (b) if the braking claim is false, then B and C will drive across and past A's position, which reveals a trajectory inconsistency. The key observation we make here is that the position and speed values in the heartbeat data sent by A, B, and C suffice to indicate which of the implied conditions in (a) or (b) holds, thus allowing D and E to accordingly conclude whether A's claim was true or false by the time C arrives in A's position. Most likely, by that time, D and E have enough reaction time to prevent an accident, and the likelihood of that is increased by the fact that no additional messages need to be exchanged between the parties (assuming that B and C, the majority in this example, send honestly computed heartbeat data; we have generalized the example to higher honesty thresholds, omitted here). In our analysis, we have measured D's reaction time (see left histogram in Figure 2 for the simulated data analysis and left histogram in Figure 3 for the real data analysis) as the inter-arrival time between vehicles C and D at A's claimed braking position. Similarly, we have measured E's reaction time (see right histogram in Figure 2 for the simulated data analysis and right histogram in Figure 3 for the real data analysis) as the inter-arrival time between vehicles C and D at A's claimed braking position.

Our simulation was generated, using Simulation of Urban Mobility [15] software, as follows. First of all, 504 vehicles are initially randomly placed in a 0.9 times 0.9 miles area in a typical urban environment in Chicago, IL, with two-way traffic roads. Each vehicle moves according to a car-following mobility model, as specified in the software used, and turns left, turns right or goes straight at an intersection with equal probability. The maximum speed

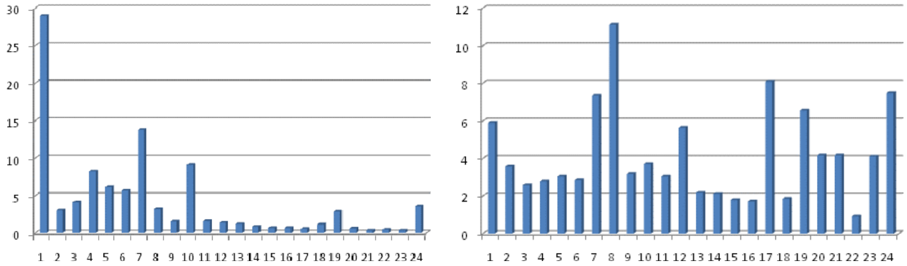

Figure 2 Histogram of vehicle D's (left) and vehicle E's (right) reaction times in 24 time slots (using simulated data).

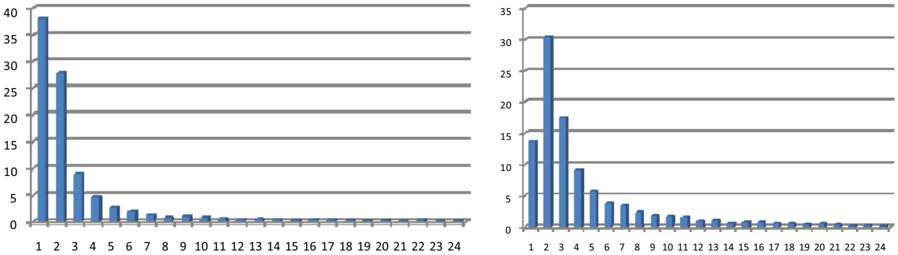

Figure 3 Histogram of vehicle D's (left) and vehicle E's (right) reaction times in 24 time slots (using real-life data).

is about 35 mile/hour, the average speed is about 15 mile/hour; and the total simulation time is 2000 seconds. In this time, a random vehicle is chosen as generating a false braking alert (i.e., vehicle A), resulting in about 1700 non-interactive elections, where the reaction time of vehicles D and E (see Figure 1) is measured.

In the left part of Figure 2 an histogram of the reaction time of vehicle D is shown, where y is the percentage of elections where D's reaction time fell into the x-th time slot, for $x = 1, \ldots, 24$, where the time interval containing 99% of the data was divided into 24 equal-size time slots. Under the rather conservative assumption that the running time of the election algorithm is completed at the end of the first time slot, we derive that vehicle D has a 71% chance of successfully detecting and ignoring A's false braking alert. In the right part of Figure 2 we show the analogue histogram for vehicle E, from which we derive that vehicle E has a 94% chance of successfully ignoring A's false braking alert.

Our real-life data analysis was based on the traffic data collected in [19] from a multi-lane highway in Pasadena, CA, at one fixed monitoring station on Nov 11, 2010. As before, the time interval containing 99% of the data was divided into 24 equal-size time slots, and under the same assumption that the election algorithm's running time is completed by the end of the first time slot, we derive that vehicle D and E have a 62% and 86% chance, respectively, of successfully detecting A's false braking claim.

For both analysis, our conservative assumptions included time estimates on the running time that vehicles take to cryptographically verify that the received heartbeat messages were sent by authorized, non-revoked, vehicle OBEs. These assumptions can be weakened by considering results and analysis from [5, 12], possibly implying improved reaction time results. Simulation variants based on different parameters and modeling approaches, including alternative honesty thresholds, mobility models, and statistic measures, provided similarly positive reaction time results.

6 Conclusions

Traffic-related malicious behavior in vehicular networks is especially worth investigating because of its very undesirable consequences (i.e., fatal accidents). Protecting against traffic-related malicious behavior is very challenging as, for instance, it may have short time duration and it may involve only a small number of vehicles whose communication is challenged by high mobility and radio constraints. Some previous work provided elegant protocols based on voting or consensus techniques to evaluate the surrounding driving scenario. In this paper we show that interaction is not necessary: voting algorithms can be implemented non-interactively by a vehicle, based on the already exchanged vehicular network communication. This has a significant positive impact on improving vehicle reaction time in the presence of these attacks. More generally, we proposed a framework for investigating traffic-related threats and efficiently securing vehicular networks against them, a highly desirable feature of future networks. More work remains to be performed to provide efficient solutions against all concrete instances of these attacks.

Acknowledgement

We thank Greg Merrit at Berkeley Highway Lab for releasing traffic data collected within their research efforts. Part of this work was performed under the U.S. Vehicle Infrastructure Integration (VII) research initiative, funded by U.S.

Department Of Transportation. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

[1] http://ivc.ep.ch/

[2] http://en.wikipedia.org/wiki/Vehicle-infrastructure-integration

[3] X. Lin and R. Lu, A bibliography on secure vehicular communications, http://bbcr.uwaterloo.ca/~rxlu/sevecombib.htm

[4] Z. Cao, J. Kong, U. Lee, M. Gerla and Z. Chen, Proof-of-relevance: Filtering False data via authentic consensus in vehicle ad-hoc networks, in *Proc. of IEEE Infocom Workshops*, 2008.

[5] J. Choi, M. Jakobsson and S. Wetzel, Balancing auditability and privacy in vehicular networks, in *Proc. of 1st ACM Workshop on Quality of Service and Security in Wireless and Mobile Networks*, 2005.

[6] G. Di Crescenzo and T. Zhang, Efficient CRL search in vehicular network PKIs, in *Proc. of the 6th ACM CCS Workshop on Digital Identity Management*, 2010.

[7] G. Di Crescenzo, S. Pietrowicz and T. Zhang, Anonymity notions for public-key infrastructure in vehicular networks, in *Proc. of Mobile Ad-Hoc Networks Sensors and Systems (MASS 07 Workshops)*, 2007.

[8] P. Golle, D. Greene and J. Staddon, Detecting and correcting malicious data in vanets, in *Proc. of ACM VANET Workshop*, 2004.

[9] J. P. Hubaux, S. Capkun and J. Luo, The security and privacy of smart vehicles, in *IEEE Security & Privacy*, 2(3): 2004.

[10] F. Kargl *et al. Secure Vehicular Communication Systems: Implementation, Performance, and Research Challenges*. CoRR, abs/0912.5393, 2009.

[11] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho and X. S. Shen. Security in vehicular ad hoc networks. *IEEE Communications Magazine*, 2008.

[12] B. Ostermaier, F. Dotzer and M. Strassberger, Enhancing the security of local danger warnings in vanets — a simulative analysis of voting schemes, in *Proc. of IEEE Conference on Availability, Reliability and Security (ARES'07)*, 2007.

[13] J. Petit and Z. Mammeri, Impact of message authentication on braking distance in vehicular networks, in *Proc. of 5th ERST^2 Workshop*, 2010.

[14] M. Raya, P. Papadimitratos, I. Aad, D. Jungels and J.P. Hubaux, Eviction of misbehaving and faulty nodes in vehicular networks, in *IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks*, 2007.

[15] Y. Kondareddy, G. Di Crescenzo and P. Agrawal, Analysis of Certificate Revocation List Distribution Protocols for Vehicular Networks. GLOBECOM 2010: 1-5.

[16] http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Main_Page

[17] M. Gerla and L. Kleinrock, Vehicular networks and the future of the mobile internet, in *Computer Networks (2011)*, doi:10.1016/j.comnet.2010.10.015.

[18] G. Di Crescenzo, Y. Ling, S. Pietrowicz and T. Zhang, Non-interactive malicious behavior detection in vehicular networks, in *Proc. of 2nd IEEE Vehicular Networking Conference (VNC 10)*, Dec. 2010.

[19] http://bhl.calccit.org/

Biography

Giovanni Di Crescenzo is a senior scientist at Telcordia Technologies with 15+ year research experience. He received a Ph.D. from University of California San Diego, USA (with a thesis on cryptography) and a Ph.D. from the University of Naples, Italy (with a thesis on zero-knowledge proofs). His main research activity has been in various areas of Mathematics and Computer Science, including Cryptography, Computer/Network/Information Security, Computational Complexity, Algorithms, and Statistics, where he has produced 100+ scientific publications in major, refereed conferences and journals, including 1 book, 2 book chapters, 3 proceedings of conferences or workshops for which he was a technical program chair.

He was awarded more than 15 among prizes and patents, including recognitions as best paper co-author and as a person of extraordinary ability in the field of science. He regularly gives invited talks and referees papers for the major conferences and journals in his areas of expertise, and has been involved in several research projects funded by government agencies (including DARPA, ARL, ARDA, NSA, DOT, IARPA). He also frequently consults for commercial institutions in the area of cryptography, security, telecommunication, telematics and intellectual property.

Yibei Ling (M'00–SM'06) received the B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1982, the M.S. degree in statistics from Shanghai Medical University (currently Fuda University), Shanghai, China, in 1988, and the Ph.D. degree in computer science from Florida State University at Miami, Miami, in 1995. He is a Senior Research Scientist with the Applied Research Laboratories, Telcordia Technologies (formerly Bellcore), Piscataway, NJ. His research interests include distributed computing, query optimization in database management system, scheduling, biological moduling, checkpointing, system performance/evaluation, fault localization/self-healing in mobile adhoc networks, and power-aware routing in mobile adhoc networks. He is the Architect, as well as the Developer, of the voice subsystem of Telcordia Notification System. He is a key team member of the Telcordia led DARPA Adaptive Cognitive-Enhanced Radio Team Project, in which he is responsible for designing/implementing the distributed positioning subsystem using GPS and ultrawideband technologies. He is an Invited Reviewer for Mathematical Reviews. Dr. Ling has published several papers in the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, and he is involved in prestigious conferences such as the Special Interest Group on Management of Data, the International Conference on Data Engineering, the International Conference on Distributed Computing Systems, and Principles of Distributed Computing. He served as a TPC member for QShine in 2005, 2006, and 2007 and for the Computer and Network Security Symposium of the IEEE International Wireless Communication and Mobile Computing Conference in 2006.

Tao Zhang received his Ph.D. from the University of Massachusetts at Amherst, U.S., in 1993. He is Chief Scientist and Director of Wireless and Vehicular Networks at Telcordia Technologies, where he directs research and advanced development in vehicular networks and next-generation wireless systems. Dr. Zhang is a Fellow of the IEEE. He holds 26 U.S. patents, coauthored the book "IP-Based Next Generation Wireless Networks" published by John Wiley & Sons in 2004 and the book "Vehicle Safety Communications: Protocols, Security, and Privacy" to be published by John Wiley & Sons in early 2012. He is Vice Chair of the IEEE ComSoc Technical Committee on Vehicular Networks and Telematics Applications and a member of the IEEE ComSoc Ad-Hoc Committee on Cloud Communications & Networking. He is serving, or has served, on editorial boards or as a guest editor for several international journals including the IEEE TVT, IEEE JSAC, IEEE ComSoc Technology News, and the Journal of Wireless

Networks. He serves on the Industry Adversary Boards for several research institutes and programs. He has been an Adjunct Professor at Stevens Institute of Technology in New Jersey, the Polytechnic University of New York, and the National Tsing Hua University in Taiwan. He was an Assistant Professor at the Beijing Jiaotong University (then Northern Jiaotong University) in Beijing, China, from 1987 to 1989.

Stan Pietrowicz is a Senior Principal Security Consultant in the cyber security practice of Applied Communication Sciences. With over 20 years of combined experience in the Communications, Intelligent Transportation, and Smart Energy sectors, Stan is responsible for business development, project management, technical delivery and research. Working for Bellcore/Telcordia and now Applied Communication Sciences 1990, Stan's present focus is Smart Grid security and operations where he pioneered the security assessment of Advanced Meter Infrastructures and received the Telcordia's first CEO Award for Innovation in 2011. Stan also participates in security research for Intelligent Transportation Systems and vehicle communications. Stan received his MSEE from Rutgers University and BEEE from Stevens Institute of Technology.