P^{2}DAP – Sybil Attacks Detection in Vehicular Ad Hoc Networks

Tong Zhou, Romit Roy Choudhury, Peng Ning, and Krishnendu Chakrabarty

***Abstract*—Vehicular ad hoc networks (VANETs) are being increasingly advocated for traffic control, accident avoidance, and management of parking lots and public areas. Security and privacy are two major concerns in VANETs. Unfortunately, in VANETs, most privacy-preserving schemes are vulnerable to Sybil attacks, whereby a malicious user can pretend to be multiple (other) vehicles. In this paper, we present a lightweight and scalable protocol to detect Sybil attacks. In this protocol, a malicious user pretending to be multiple (other) vehicles can be detected in a distributed manner through passive overhearing by s set of fixed nodes called road-side boxes (RSBs). The detection of Sybil attacks in this manner does not require any vehicle in the network to disclose its identity; hence privacy is preserved at all times. Simulation results are presented for a realistic test case to highlight the overhead for a centralized authority such as the DMV, the false alarm rate, and the detection latency. The results also quantify the inherent trade-off between security, i.e., the detection of Sybil attacks and detection latency, and the privacy provided to the vehicles in the network. From the results, we see our scheme being able to detect Sybil attacks at low overhead and delay, while preserving privacy of vehicles.**

***Index Terms*—Coarse-grained hash, fine-grained hash, privacy, security, Sybil attack, vehicular ad hoc network**

I. Introduction

VEHICULAR Ad hoc Networks (VANETs) are being advocated as a means to increase road safety and driving comfort, as well as to facilitate traffic control [1] [2]. For example, cars can collectively sense information about traffic congestion and relay them to other cars, toll stations, or the Department of Motor Vehicle (DMV) to facilitate traffic re-routing. Several other applications can become feasible if vehicles cooperate among themselves to achieve a common goal. When designing a cooperation-based system, it is important to address security and privacy concerns. The system needs to be robust to non-cooperating entities, and should ideally be able to detect/punish them quickly. To ensure the authenticity of messages propagated in VANET, a straight-forward method is to use public keys certified by a certification authority (CA) to sign the messages. The certified public keys are called "pseudonyms". On the other hand, in order to prevent vehicles from being tracked by identifying the keys that are used, each vehicle can switch among multiple pseudonyms, which are difficult to correlate to each other. With this approach, it is difficult for an attacker to identify vehicles by examining the used keys. The above scheme has been proposed by many researchers [3], [4], [5], [6], [7], and works efficiently.

Although the above method protects the privacy of the vehicles, it leaves another security hole. Since it is difficult to tell whether two messages are from the same vehicle by examining their public keys, a malicious vehicle may pretend to be multiple vehicles (*a Sybil attack*) and then distribute false information. The deleterious effects of such attacks can cascade through the network. Vehicles are expected to obtain a new pseudonym from a trusted Road-Side Box (RSB) (serving as a CA) immediately before the earlier pseudonym expires. In [8] and [9], a light-weight solution is proposed to solve this issue. Vehicles only hold one valid pseudonym at a time, and are expected to obtain a new pseudonym from a trusted Road-Side Box (RSB) or from the on-line CA if the current pseudonym becomes invalid. In this scheme, it is critical that the vehicles have access to a CA when it needs to update its pseudonym. Without such an online infrastructure support, the vehicles are not able to obtain new pseudonyms and send signed messages. Also, if an attacker compromises an RSB, he can issue many certified pseudonyms to malicious vehicles, thus creating false messages in that area.

Yet another technique exploits directional antennas to identify the position/direction from which a message arrives [10]. A car launching a Sybil attack is expected to get caught because all the duplicate messages will come from the same position. However, in dense networks, localization errors can lead to frequent false positives. More importantly, a smart attacker may use directional antennas to mislead its neighbors about its directions.

In this paper, we propose a privacy-preserving scheme to detect Sybil attacks in VANETs under a commonly used framework in the existing work [11]. The framework assumes that vehicles communicate with each other in a multihop manner, and the communication is monitored by an RSB through passive overhearing. The RSB is securely connected to the DMV via a backhaul wired network. The DMV plays the role of a certificate authority (CA), and has the ability to manage vehicle registration, ownership, and other administrative policies. Our scheme requires the DMV to provide vehicles with a pool of pseudonyms that are used for hiding the vehicle's unique identity. On the other hand, to prevent a vehicle from using multiple pseudonyms to direct a Sybil attack, the pseudonyms assigned to a particular vehicle are hashed to a common value. By calculating the hashed values

Manuscript received 5 January 2010; revised 7 May and 12 July 2010. A preliminary version of this paper appeared in *Proc. of the 2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems, Vol. 00.*

T. Zhou, R. R. Choudhury, and K. Chakrabarty are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 (e-mail: tzhou@ee.duke.edu, romit@ee.duke.edu, krish@ee.duke.edu).

P. Ning is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 (e-mail:pning@ncsu.edu).

Digital Object Identifier 10.1109/JSAC.2011.110308.

of the overheard pseudonyms, an RSB and the DMV will be able to determine whether the pseudonyms came from the same pool, thus helping to identify a Sybil attack. In this scheme, privacy is preserved as long as the RSB can be trusted. However, a compromised RSB may be able to "single out" a vehicle by assimilating all the pseudonyms hashing to a unique value. Several other challenges arise while attempting to incorporate both privacy and security into a vehicular network system. We discuss these challenges, and address them systematically through a light-weight, scalable protocol that we call "Privacy Preserving Detection of Abuses of Pseudonyms" (P^2DAP). The details of P^2DAP are presented in Section III. Though our algorithm requires the calculation/storage of the pseudonyms, we prove that the computational overhead and storage overhead are affordable.

The rest of the paper is organized as follows. Section II describes the system model and assumptions. Section III presents the proposed detection schemes to handle specific requirements. Section IV further discusses the P^2DAP algorithm and proposes three possible improvements. Section V presents simulation results. Finally, Section VI concludes the paper and outlines future research directions.

II. System Model

In this section, we describe our assumptions regarding to the VANET system, capabilities of the attackers, and the Sybil attacks.

A. Assumptions on VANET Architecture

1) The *DMV* is the trusted party that maintains vehicle records and distributes certified pseudonyms to vehicles. The DMV has enough resources to generate pseudonyms quickly and store all the vehicle-related information, and is referred to when any authoritative clarification is necessary. However, such DMV services are not designed for heavy network traffic – excessive communication can cause the DMV to become a bottleneck.
2) *Vehicles* are untrusted parties. They communicate with each other in a multihop manner. A message exchanged among vehicles is signed with a DMV-certified pseudonym.
3) *RSBs* are wireless access points. They are scattered along the road and connected to the DMV via a backhaul network, acting as intermediates to the DMV. The RSBs monitor vehicular activity, identify suspicious behavior, and report to the DMV for confirmation and punishment. The RSBs may be compromised, thus they cannot be used for critical functions – for example, the RSB cannot authenticate a message or distribute pseudonyms. However, they can be used to improve the scalability of a system.

The overall architecture of a VANET, is shown in Figure 1.

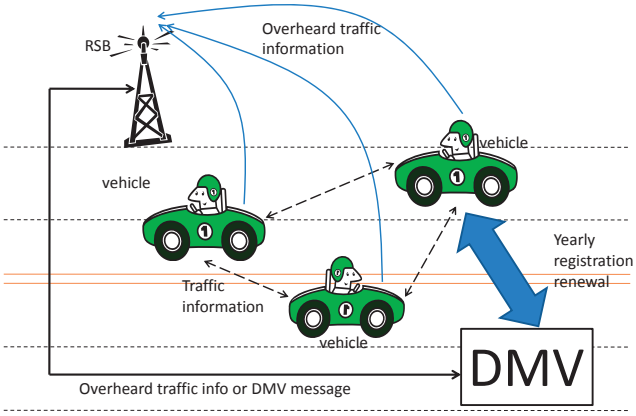


Fig. 1. The architecture of VANET.

B. Assumptions of Attackers' Actions

In this section, we discuss the actions of attackers that we are interested in.

1) *Announce false messages – False data injection:* A vehicle can sign a false message and then broadcast it. Such an attack cannot be detected from the message itself, since the message can be signed by a CA-certified pseudonym. This attack may be addressed by a majority voting scheme if there are more benign vehicles than attackers. However, the voting scheme fails if the attacker carries out Sybil attack by generating sufficient false identities to outvote benign vehicles, which we will discuss next.
2) *Pretend to be multiple vehicles by using multiple pseudonyms – Sybil attack:* In VANET, a vehicle can carry out a Sybil attack by using its multiple pseudonyms to sign messages. If a vehicle can be identified from a set of pseudonyms it uses, then the vehicle's privacy is compromised. As a result, vehicles and RSBs that overhear multiple messages signed with an attacker's pseudonyms have no means of recognizing that these pseudonyms actually belong to the same vehicle. This paper proposes a method to solve this problem.
3) *Compromised RSBs:* RSBs are semi-trusted parties, and some RSBs may be compromised by the attackers. We assume that RSB compromises can be easily detected by the DMV, thus being quickly revoked. However, attackers can still gain information stored in the RSBs. Therefore, a scheme's resilience to RSB compromise is determined by the amount of information released to and held by the RSBs.

In summary, it is necessary to have a framework allowing RSBs to detect a Sybil attack, without knowing the association between pseudonyms and unique vehicle IDs (i.e., without compromising privacy). The framework design needs to be scalable in terms of the workload it imposes on the DMV, and needs to be robust against RSB compromise. Moreover, it is important to quickly detect the attack and perform subsequent punishments/revocations to minimize the impact of the attack. To address these concerns, we propose a new scheme, referred as Privacy-Preserving Detection of Abuses of Pseudonyms (P^2DAP).

C. Structure of Events and the Use of Pseudonyms

In vehicular network applications, vehicles are expected to broadcast specific events whenever they observe them.

Counting the number of vehicles that send the same message is an important primitive that several VANET applications depend on. To achieve the notion of *same* or *different* events, we need to unambiguously define the format of "events".

An event is a tuple (t, l, e) *generated at a pre-defined time interval* $t \in \mathbf{T}$*, in a pre-defined region* $l \in \mathbf{L}$ *for an event type* $e \in \mathbf{E}$*, where* $\mathbf{T}, \mathbf{L}, \mathbf{E}$ *are distributed to vehicles.*

An attacker carries out the Sybil attack by abusing multiple pseudonyms. On the other hand, in order to avoid being tracked, benign vehicles can also use multiple pseudonyms to report events. In order to distinguish the benign use of pseudonyms from the abuse of pseudonyms, we now introduce the following restriction on the use of pseudonyms.

A benign vehicle can use only one pseudonym to sign one event.

If a vehicle uses multiple pseudonyms to sign an event such that others think there are multiple vehicles reporting the same event, the action is considered to be a *Sybil attack*, and the vehicle is deemed to be malicious.

III. The Proposed P^{2}DAP Scheme

This section presents our scheme on handling Sybil attacks. The main purpose is to detect Sybil attacks and revoke malicious vehicles immediately after detection. A baseline method is to forward all the reported events to the DMV, and let the DMV examine the signatures of each message. On observing a single event (t_i, l_j, e_k) signed with two different pseudonyms of the same vehicle, the DMV considers that vehicle as an attacker. The drawback of this method is the heavy network traffic on the DMV. Therefore, we propose P^{2}DAP schemes in which RSBs perform most of the DMV's task to reduce the communication overhead. We also discuss how our schemes preserve privacy in case of RSB compromise.

A. *Complete Two-Stage P^{2}DAP Scheme*

We first propose the Complete Two-Stage P^{2}DAP Scheme, abbreviated as C-P^{2}DAP. Later, we will propose a number of variances of C-P^{2}DAP to improve the performance of the scheme. In P^{2}DAP scheme, we delegate most of the detection to RSBs, and involve the DMV only when suspected vehicles need to be confirmed as a Sybil attacker. However, since RSBs are not trusted entities, the vehicle information available to the DMV cannot be transferred to the RSBs. In view of these constraints, we divide the vehicles into groups, and release the group information to RSBs. Such information allows RSBs to detect suspicious behavior, but is not sufficient for RSBs to track vehicles, because RSBs cannot distinguish a vehicle from a group of vehicles. To group the vehicles, we use the one-way hash function to hash the pseudonyms during initialization.

Initialization Step

Initially, the DMV knows the total number of vehicles, and sequentially generates a sufficient number of yearly pseudonyms for all the vehicles. After generating a pseudonym p, the DMV first hashes $(p \mid \kappa_c)$ using a one-way hash function, where κ_c is a global key. It then selects a set of bits from the hashed result to create hash collisions. The selected bits are referred as "coarse-grained hash value". After that, the pseudonym p is placed into a group, which stores the pseudonyms with the same coarse-grained hash values. In other words, for each pseudonym p_l in the m-th coarse-grained group, we have $H(p_l|\kappa_c) = \Gamma_m$, where H is a one-way hash function, and Γ_m is the coarse-grained hash value for group m. We refer such groups as "coarse-grained groups". The key κ_c will be distributed to all the RSBs.

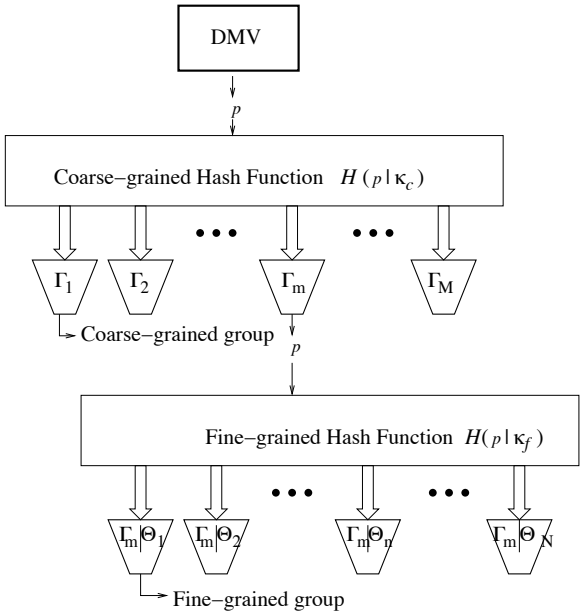


Fig. 2. The generation and two-level hashing of a pseudonym p.

Next, the DMV calculates the hash value for the above p with a new key κ_f, and selects a set of bits from the result. The bits selected from the new hash value are referred as the "fine-grained hash value". The pseudonym p is then placed into a subgroup of the coarse-grained group, namely fine-grained group, in which all the pseudonyms have the same fine-grained hash value. For each pseudonym $p_{l'}$ in the n-th fine-grained group under the m-th coarse-grained group, we have $H(p_{l'}|\kappa_f) = \Theta_n$, where Θ_n is the fine-grained hash value for the subgroup n.

The above steps are referred as "two-level hash", and are shown in Figure 2. The DMV keeps generating and two-level hashing pseudonyms until all fine-grained groups contain enough pseudonyms for a vehicle's use. After that, the DMV loads a unique fine-grained group of pseudonyms to each vehicle at the time of yearly vehicle registration, and stores the corresponding $(\Gamma_m|\Theta_n)$ as the vehicle's secure plate number. From the above description, it is obvious that the mapping from secure plate numbers to vehicles is one-to-one. Thus, the DMV needs to carefully choose the length of Γ_m and Θ_n, such that the total number of available secure plate numbers are greater than or equal to the number of vehicles.

The two-level hashing saves storage for the DMV, because the DMV can link a pseudonym to a vehicle by calculating its coarse-grained and fine-grained hash values, and then comparing them with the secure plate number $(\Gamma_m|\Theta_n)$. This obviates the need of maintaining vehicle secure plate numbers and pseudonym association.

After the initialization stage, the DMV stores the secure plate number for each vehicle, and secretly keeps the fine-grained hash key κ_f.

Generating Pseudonyms with Short-Period Keys

When generating the pseudonyms, we need to consider the lifetime of a coarse-grained key κ_c, because an attacker gaining access to an RSB can partially learn the pseudonyms of all the vehicles for that lifetime. If the lifetime is too long, the privacy of the vehicles will be severely impaired. Therefore, κ_c should be given a short lifetime such as one or two days. We can then modify the initialization stage as follows.

We divide a whole year's time into Ω time intervals. Each time interval can be one day. In the initial pseudonym generation, the DMV uses a set of coarse-grained keys K_c, instead of one key, κ_c, to hash the pseudonyms. Each key $\kappa_{c,\gamma} \in K_c$ is used to generate pseudonyms for the γ-th time interval. The pseudonyms that are hashed to Γ_m with the key $\kappa_{c,\gamma}$ will be put into the m-th coarse-grained group, and they can only be used in the γ-th time interval. After that time interval, these pseudonyms will be discarded. To prevent malicious vehicles from using expired pseudonyms to sign events, the DMV uses different keys to generate certificates for pseudonyms in different time intervals. Thus, vehicles can recognize an expired pseudonym by examining its certificate. Initially, the DMV secretly holds all the coarse-grained hash keys. At the beginning of γ-th time interval, the DMV releases the key $\kappa_{c,\gamma}$ to the RSBs. By this approach, an RSB holds each valid coarse-grained key only for a short time. When an RSB is compromised, the attacker only obtains the coarse-grained hash key for the current time interval. We do not impose any restrictions on the fine-grained key κ_f, because the DMV does not release it, and an attacker cannot obtain it by compromising an RSB.

Comparing to the long-period keys, this short-period key generation uses Ω coarse-grained hash keys instead of one, thus bringing an extra storage overhead to the DMV. At the same time, the computational overhead will increase, which we will show in Section V-B.

Sybil Attack Detection Step

When vehicles communicate, an RSB overhears all the vehicles within their communication range, and puts the pseudonyms used to sign the event (t_i, l_j, e_k) in the list $L_{i,j,k}$. When all pseudonyms for the event (t_i, l_j, e_k) are collected, the RSB detects the Sybil attacks as follows – The RSB goes through each pseudonym p in the list $L_{i,j,k}$ and computes the coarse-grained hash value $H(p|\kappa_c)$. (Recall that κ_c is pre-distributed to all RSBs in the initialization step.) If $\exists p, p' \in L_{i,j,k}$ such that $H(p|\kappa_c) = H(p'|\kappa_c)$, then the RSB notices that two pseudonyms of the same coarse-grained hash value are used to sign the event (t_i, l_j, e_k). This can be either (i) a Sybil attack where one vehicle is using multiple pseudonyms to report the same event, or (ii) a *false alarm*, where an event is reported by two vehicles whose pseudonyms are in the same coarse-grained group. The RSB cannot discriminate between (i) and (ii) and it sends the report to the DMV. The report contains the event (t_i, l_j, e_k), the hash value Γ, the pseudonyms whose coarse-grained hash value is Γ, the signatures of the event, and the certificates accompanying the pseudonyms.

On receiving an RSB report, the DMV first verifies the signatures and the coarse-grained hash value Γ to prevent a compromised RSB from implicating a benign vehicle. If the RSB proves to be bonafide, the DMV calculates the fine-grained hash value $H(p|k_f)$ for each pseudonym p in the RSB report. If $\exists p, p'$ in the report such that $H(p|\kappa_f) = H(p'|\kappa_f)$, the DMV concludes that p and p' are from the same vehicle that has attempted a Sybil attack. The DMV then takes further action to punish or revoke the malicious vehicle.

In this scheme, a Sybil attack is guaranteed to be detected. However, when the vehicles are densely distributed, false alarms can happen often.

B. E-P^{2}DAP – Detecting Events Instead of Sybil Attack

In the C-P^{2}DAP scheme, an RSB reports to the DMV whenever it finds any set of pseudonyms that hash to the same coarse-grained values. Thus, when an event is reported by a large number of vehicles, C-P^{2}DAP can cause false alarms. Clearly, on a road with heavy traffic, such false alarms will create a heavy communication overhead on DMV. To address this problem, we observe that detecting each and every Sybil attack may not be necessary for practical VANET applications. We first make the following assumptions: (i) each false (faked) event is generated by only one malicious vehicle; (ii) benign vehicles will not report false events. We then propose the Event-P^{2}DAP scheme (abbreviated as E-P^{2}DAP), which does not detect all Sybil attacks, but only detects those creating false events.

For an event (t_i, l_j, e_k), the RSB collects a list of pseudonyms $L_{i,j,k}$ used to sign the event. If $\forall p, p' \in L_{i,j,k}$, $H(p|\kappa_c) = H(p'|\kappa_c)$, i.e., all the pseudonyms used to sign (t_i, l_j, e_k) have the same coarse-grained hash value, then the event is probably sent from only one vehicle, and is likely a faked event. In this case, the RSB generates a report with the same format as in C-P^{2}DAP and sends it to the DMV.

In this scheme, the DMV only needs to examine the pseudonyms in two cases: 1) an attacker reports a false event and carries out a Sybil attack; 2) a true event is reported by multiple benign vehicles whose pseudonyms have the same coarse-grained hash value, which is a false alarm. Obviously, the number of false alarms is likely to be small compared to the total number of the pseudonyms that RSBs process. Therefore, the RSBs are able to efficiently take over most of the pseudonym processing tasks, thus reducing the burden on the DMV.

C. T-P^{2}DAP – Detecting Collusion

One issue with the E-P^{2}DAP scheme is that it cannot detect colluding vehicles, i.e., two or more malicious vehicles reporting the same faked event. In order to address the collusion, we propose Threshold-P^{2}DAP (abbreviated as T-P^{2}DAP), described as follows – We assume each faked event is generated by a small number of colluding attackers instead of one attacker, but that number will not exceed a threshold τ. Then, for a pseudonym list $L_{i,j,m}$, the RSB calculates the coarse-grained hash value for each pseudonym $p \in L_{i,j,k}$, and obtains a set of coarse-grained hash values S_c. If $|S_c| \leq \tau$ and two or more pseudonyms in $L_{i,j,k}$ map to the same coarse-grained hash value, the RSB suspects the event to be fake and reports to the DMV. Similar to the above C-P^{2}DAP and

E-P^{2}DAP schemes, the DMV in the T-P^{2}DAP scheme then examines the RSB report and finds out whether the event is from attackers.

Comparing to E-P^{2}DAP, T-P^{2}DAP is more resilient to collusion. Any false event reported by less than τ attackers can be detected by an RSB. Obviously, T-P^{2}DAP has a larger false alarm rate than E-P^{2}DAP.

In this section, we introduced C-P^{2}DAP, E-P^{2}DAP, and T-P^{2}DAP. Note that the first 2 schemes are special cases of T-P^{2}DAP. In T-P^{2}DAP, if $\tau = 1$, an RSB detects a Sybil attack by verifying that the coarse-grained hash values of the pseudonyms used to sign a single event are the same. Thus, T-P^{2}DAP becomes E-P^{2}DAP. On the other hand, when $\tau \geq$ number of coarse-grained hash values, an RSB reports a Sybil attack once it finds the coarse-grained hash values of two pseudonyms used to sign an event are the same. In this case, T-P^{2}DAP becomes C-P^{2}DAP. Currently, the T-P^{2}DAP cannot detect the colluding vehicles if each malicious vehicle only reports a faked event with one pseudonym. However, such an attack is not a Sybil attack and is beyond the scope of this paper. Section V will show more details of performance comparison between the three schemes.

IV. Discussions: Improvements on P^{2}DAP

In this section, we propose several improvements to the above P^{2}DAP schemes from the perspectives of key revoking convenience and adaptivity.

A. *Revoking the Pseudonyms of Malicious Vehicles*

After a malicious vehicle is detected, the DMV should revoke all its pseudonyms. In this subsection, we discuss three possible approaches of revoking vehicles that can be combined with P^{2}DAP.

The first two approaches are to use the revocation schemes proposed in [12], i.e., Revocation of the Tamper-Proof Device (RTPD) and Revocation using Compressed Certificate Revocation Lists (RC^{2}RL). The RTPD requires the hardware-support on the vehicle, in which a tamper-proof device (TPD) used to store pseudonyms and sign messages is installed on each vehicle. On observing a malicious vehicle, the DMV sends a revocation message to the TPD, then the TPD will erase all the pseudonyms and stop signing messages. By this approach, the DMV can revoke a vehicle in a single message.

Different from RTPD, RC^{2}RL does not assume any hardware support such as the TPD; Instead, it creates a bloom filter for all the pseudonyms to be revoked. The bloom filter is then broadcasted to all the vehicles. On receiving a message, a vehicle uses the bloom filter to verify its pseudonym, and drop it if the pseudonym is found revoked In the paper [12], the size of each bloom filter is estimated as tens of Kbytes. Therefore, in order to revoke a vehicle, the DMV needs to flood tens of Kbytes throughout the network.

The third approach is to create a secret key for each vehicle that helps to identify its pseudonyms. Such a secret key is regarded as a "backdoor" in this paper. The Group Signature (GS) schemes proposed in [6] can be used to generate such a backdoor. In the GS scheme, each vehicle is equipped with a group public key gpk_{CA} and a private signature key gsk_V, and generates its own pseudonyms. On creating a pseudonym K_V^i, the vehicle generates a group signature $\Sigma_{CA,V}(K_V^i)$ for K_V^i with the private key gsk_V, and then calculates a certificate $Cert_{CA}^H(K_V^i)$ using the group public key gpk_{CA}. Other vehicles can verify K_V^i by validating $\Sigma_{CA,V}(K_V^i)$ using $Cert_{CA}^H(K_V^i)$. When the DMV needs to revoke a vehicle, it simply broadcasts the vehicle's private signature key gsk_V.

The idea of GS is adopted in the initialization stage of P^{2}DAP, in which the DMV generates gsk_V and $\Sigma_{CA,V}(K_V^i)$ for the vehicle V when generating a pseudonym K_V^i. The DMV will distribute the pseudonyms, the group signatures, and the group public key gpk_{CA} to the vehicles, and keep the private key gsk_V as the "backdoor" of V. The vehicles can then use the pseudonyms to sign the messages, and use gpk_{CA} to verify certificates of the pseudonyms. When the DMV needs to revoke a vehicle V, it broadcasts gsk_V, like the DMV does in the GS signature scheme. On receiving gsk_V, a vehicle can verify whether a pseudonym is from vehicle V, thus being able to identify the messages from a revoked vehicle.

In this subsection, three different approaches of revoking pseudonyms are discussed. In these approaches, RTPD incurs small communication and computation overhead, but it requires hardware support. RC^{2}RL and GS do not need hardware support, but they require large communication overhead and computation overhead, respectively. Each method has its pros and cons, and which one to choose depends on the system's available resources.

B. *τ-P^{2}DAP: Real-Time Adaptive P^{2}DAP Scheme*

This subsection discusses the P^{2}DAP scheme adaptive to the real-time traffic. For a particular RSB, the number of nearby vehicles is a time-varying value. For example, the study in [13] shows that within one day, the traffic volume in a street near Incheon International Airport ranges from 10 vehicles/hour to 3000 vehicles/hour. Such fluctuations in traffic volume causes difficulty in using a single detection scheme to efficiently detect the attackers. When there are fewer vehicles, the E-P^{2}DAP or the T-P^{2}DAP with a small threshold is preferred, such that the malicious vehicles can be detected with a smaller cost. When the traffic volume is high, the C-P^{2}DAP, or the T-P^{2}DAP with a large threshold, is selected to better catch collusions. A method to solve this problem is to make each RSB adaptively choose detection scheme the DMV based on the traffic volume.

We then propose τ-P^{2}DAP, in which each RSB checks the total number of received packets for each reported event (written as N_{PE}) when it attempts to detect a Sybil attack. N_{PE} can then be a parameter to calculate the value of τ in the T-P^{2}DAP, using the equation $\tau = \alpha N_{PE}$, where α is an estimated proportion of attackers among all vehicles. The value of α can be either pre-distributed to the RSBs, or learned by the RSBs during detection of attackers. Using α, we estimate the number of nearby attackers, and use this estimation as the value of τ.

With τ-P^{2}DAP, the RSB is expected to adaptively report to the DMV based on the current traffic status. The algorithm may not always be the best one if the malicious vehicles intentionally mislead the RSB. For example, the malicious

vehicles can use an extremely large number of pseudonyms to sign an event, such that the RSB is forced to use C-P^{2}DAP and incurs huge communication overhead to the DMV. Also, several colluding malicious vehicles can generate many events, each signed with a small number of pseudonyms, such that the RSB will use E-P^{2}DAP algorithm and miss these events. However, in the first case, these malicious vehicles are guaranteed to be quickly detected, and the large communication overhead hence brought up will only last for a short period. In the second case, these faked events can be easily filtered out when there is a large enough number of benign vehicles around.

C. κ-P^{2}DAP: Distribute Different Information to Different RSBs

This subsection discusses the adaptive P^{2}DAP algorithm from the location perspective. Even in the same city or county, the traffic volume varies significantly on different roads. As shown in a survey of traffic volume in 2002 at Columbia, NY [14], on different streets in the Columbia county, the average traffic volume ranges from 100 to 25,000 vehicles/day. Thus, it is difficult to determine the number of coarse-grained groups when the P^{2}DAP scheme is applied. A small number of coarse-grained groups will result in many false alarms on a highway, while a large number can harm the privacy of vehicles in a small community. To solve this issue, we propose κ-P^{2}DAP, which extends the 2-level hash keys to n-level hash keys to cope different traffic volumes. In this scheme, the DMV distributes different subsets of the hash keys to the RSBs based on the traffic volume near them.

When generating a pseudonym, instead of using two hash keys, the DMV uses q hash keys $\{\kappa_1, \cdots, \kappa_q\}$ to calculate the hash values. With each hash key κ_i, the DMV calculates an ϵ-bit hash value $\Gamma_{i,j} = H(p_j|\kappa_i)$ for the pseudonym p_j. The hash values for the pseudonym p_j is written as $\mathbf{\Gamma_j} = (\Gamma_{1,j}, \cdots, \Gamma_{q,j})$, and can be considered as an element of a q-dimensional space $\mathcal{V}$, in which each dimension has 2^ϵ elements. The DMV keeps generating pseudonyms, until their hash values fill the space $\mathcal{V}$.

After all the pseudonyms are generated, the DMV distributes the pseudonyms that can hash to the same value with the keys $\{\kappa_1, \cdots, \kappa_\beta\}$ to the same vehicle, where β is chosen such that $2^{\beta\epsilon} \geq N_V$. After that, DMV adaptively release the last δ hash keys $\{\kappa_{q-\delta+1}, \cdots, \kappa_q\}$ to the RSB, where δ is determined by the traffic volume around the RSB. If we want the RSB not being able to distinguish among vehicles passed by within an hour, we should have $\delta = \frac{\log_2 K_V}{\epsilon}$, where K_V is the number of vehicles passing by the RSB within an hour. Such an indistinguishability of one vehicle among N multiple vehicles is defined as $N-anonymity$ [15], which is an important metric of privacy. More details about anonymity will be discussed in Section V-C.

One possible issue of κ-P^{2}DAP is that an attacker can compromise an RSB that watches heavy traffic and use the knowledge to track vehicles on roads with less traffic. In this scenario, the vehicles will lose their privacy. Therefore, when releasing keys to an RSB, the DMV needs to consider both the local traffic and the cost of compromising the RSB. RSBs with more hash keys are expected to be more difficult to compromise. κ-P^{2}DAP has increased computational overhead of DMV and the RSBs. However, the communication overhead remains the same as the other P^{2}DAP schemes discussed in Section III.

TABLE I
PARAMETERS USED IN SIMULATION.

Params	Dense Vehicles	Sparse Vehicles
Street length (m)	2,000	20,000
Comm Radius (m)	200	50
Street Width (lanes)	3	3
Lane Width (m)	3	3
Vehicle Speed (m/s)	25 – 35	25 – 35
Pseudonyms/Vehicle per Day	20	20
Vehicle Packet Rate (pkts/s)	3	3
Simulation Time (s)	400	800

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of P^{2}DAP using the following metrics: computation/communication overhead of the DMV, privacy of the vehicles, and detection latency. We also examine the tradeoff between detection latency and overhead. Results from the evaluation are expected to offer insights into the design of practical vehicular networks.

A. Simulation Setup

The P^{2}DAP scheme is simulated in ns-2 version 2.29. We use the 802.11a MAC and PHY layer protocol, and the SHA-1 hashing as our hash function. In the initialization stage, we use SHA-1 hashing function to generate pseudonyms that are enough for our use. We also simulated two different scenarios – one comprised densely-distributed vehicles (in Sections V-D and V-E), and the other comprised sparsely-distributed vehicles (in Section V-F). The major simulation parameters are listed in Table I.

In our simulation, we randomly generate events with different time intervals, locations, and types, and then store them in a global array. When generating the events, we consider the length of time interval as 20 seconds, and the length of location segment as 250m, with a total number of 5 different event types. Each vehicle periodically accesses the array, obtains the events with current time interval and the vehicle's current location, and broadcasts them. A vehicle also relays events heard from other vehicles.

We have defined four types of nodes: benign vehicle, malicious vehicle (attacker), RSB and the DMV. A benign vehicle frequently senses the events, sign and broadcast them. Meanwhile, an attacker generates a random number of events, then signs each event with multiple pseudonyms and broadcasts them. Due to the small number of event types in our simulation, there is a high probability that two attackers report the same event, thus creating a colluding scenario. This behavior of attackers is called "semi-collusion", because there are times that they report events individually. We intuitively create such a behavior to test P^{2}DAP's resilience to colluding attackers.

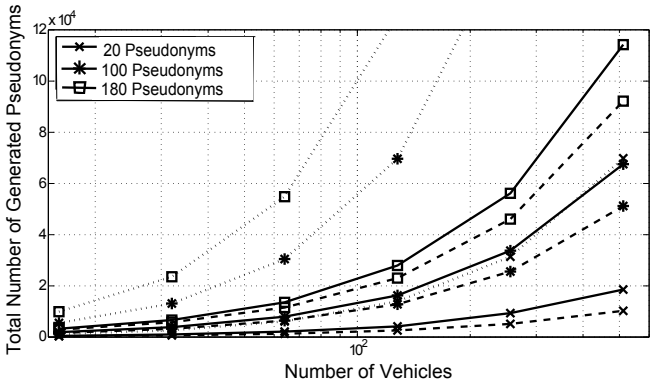


Fig. 3. Computational overhead of the DMV when generating pseudonyms. Solid lines are the simulated results, dotted lines are the theoretically-calculated upper bound values, and dashed lines are the theoretically-calculated lower bound values.

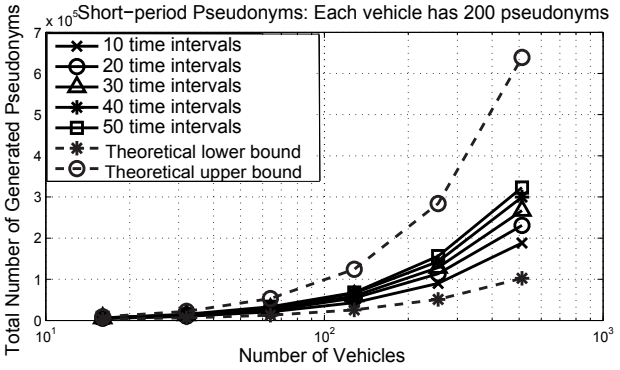


Fig. 4. Computational overhead generating short-period pseudonyms. Each vehicle has a total of 200 pseudonyms.

In the following subsections, we will present our results for the following schemes:

- C-P^{2}DAP - Detects all Sybil attacks.
- E-P^{2}DAP - Detects Sybil attacks that generate false events.
- T-P^{2}DAP - Detects collusions of a threshold number of attackers.
- τ-P^{2}DAP – Detects collusion with a traffic-volume-adaptive threshold number of attackers.

κ-P^{2}DAP only differs from other algorithms in the initialization stage, and does not have a different behavior when detecting Sybil attacks. Therefore, we do not evaluate its performance in our simulation.

B. Theoretical and Experimental Results: Computational Overhead of Generating Pseudonyms

Assume we have N_V vehicles in total, while each vehicle needs M pseudonyms. We also assume a hash function generating evenly-distributed hash values. We first calculate an upper bound (defined as N_u) of the expected number of pseudonyms that the DMV needs to generate for all vehicles. We start from the case of $M = 1$, in which the problem is converted to a *coupon collector's problem* [16]. The expected number of generated pseudonym is $N_p \equiv N_V \log N_V + \mu N_V + \frac{1}{2} + o(1)$, where $\mu \approx 0.577$. Thus, for $M > 1$, $N_u = M \times (N_V \log N_V + \mu N_V + \frac{1}{2} + o(1))$. On the other hand, we can easily find from the definition of N_p that it has the lower bound of $O(MN_V)$. Therefore, we conclude that in order to generate a year's pseudonyms, the number of pseudonyms that the DMV needs to generate is between $O(MN_V)$ and $O(MN_V \log N_V)$.

We next calculate the cost of generating short-period pseudonyms. In this scenario, the pseudonyms of each vehicle are divided into d equal portions, and each portion is hashed with a unique key. Therefore, with each hash key, the DMV needs to generate M/d pseudonyms for each vehicle. In this case, we have

$$N_u = d(\frac{M}{d} \times (N_V \log N_V + \mu N_V + \frac{1}{2} + o(1))) \quad (1)$$

$$= M \times (N_V \log N_V + \mu N_V + \frac{1}{2} + o(1)) \quad (2)$$

While the lower bound of the number of pseudonyms is still $O(MN_V)$. Therefore, the upper bound and the lower bound of the expected number of generated pseudonyms for short-period keys remains the same. Obviously, in the extreme case where each time interval only has one pseudonym, the expected number of generated pseudonyms will reach the upper-bound N_u.

We then use the simulator to generate the pseudonyms. Since we only need to obtain the number of pseudonyms generated by the DMV, we stopped the simulation right after the initialization stage. The comparison between the theoretical results and the experimental results of generating long-period pseudonyms are shown in Figure 3. It can be seen that the experimental results fall between the calculated upper bounds and lower bounds.

Also, in Figure 4, we show that short-period keys increases the expected number of generated pseudonyms. When dividing pseudonyms into 50 short periods, the number of generated pseudonyms almost doubled, yet it has not reached the upper bound N_u.

C. Experimental Results: Privacy

We first give the definition and metric of privacy in our scenario. If an RSB is compromised, the attacker can obtain the coarse-grained hash keys stored in the RSB, thus learning the coarse-grained hash values of all the pseudonyms. However, because the coarse-grained hash values are shared among multiple vehicles, the knowledge of a vehicle's coarse-grained hash value does not compromise its anonymity completely. Here we are using the k-anonymity model in [15] to evaluate privacy; in order to avoid confusing k in the k-anonymity with our keys, we rename the model of privacy as N-anonymity and apply its definition to vehicular networks:

Given a set of vehicles $\{V_i\}_{1 \le i \le N_V}$, a set of attribute values A and a one-way attribute function $F : \{V_i\} \to A$, the vehicle set is said to achieve N-anonymity if and only if for each attribute value $a \in F(\{V_i\})$, there are at least N occurrences of a in $F(\{V_i\})$, where N_V is the number of vehicles.

From this definition, we see that the anonymity of the vehicles to an RSB equals to the number of fine-grained groups in each coarse-grained group. Therefore, we conclude that the anonymity of vehicles in case of RSB compromise is

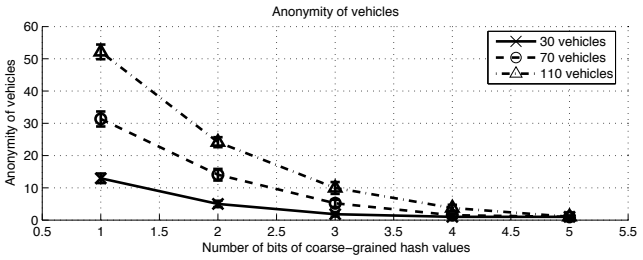


Fig. 5. Anonymity of a subset of all the vehicles

2^{n_f}, where n_f is the number of bits in the fine-grained hash value. In other words, the anonymity is $M/2^{n_c}$, where n_c is the number of bits in the coarse-grained hash value. In order to study the privacy of vehicles in a subset of all the vehicles, we generate pseudonyms for 256 vehicles, and randomly pick a subset of vehicles to examine their anonymity. The results are shown in Figure 5, from which we see the anonymity of the vehicles quickly converges to 0 when the number of bits of coarse-grained hash values goes to 5. For more vehicles, we expect a longer coarse-grained hash value is required to reduce the anonymity. For 2^{24} (more than a million) vehicles, we expect a 20-bit coarse-grained hash value can make the anonymity 0.

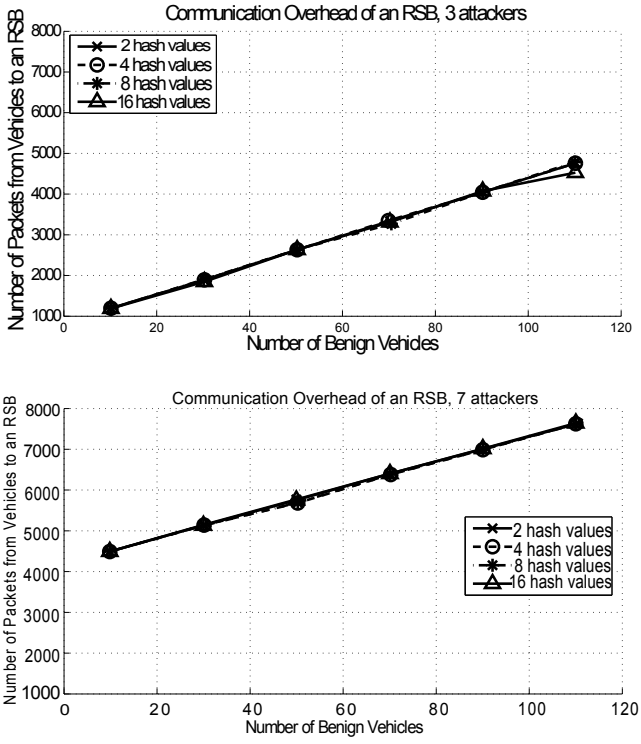


Fig. 6. Number of packets from vehicles to an RSB: for all our proposed schemes

D. Experimental Results: Communication Overhead

1) Overhead on the RSBs: Figure 6 shows the number of packets processed by an RSB. From the figure, obviously the number of packets received by an RSB increases with the increase in the number of attackers or the number of benign vehicles. We include these results here for the later comparison of the overhead on the DMV and show the reduction of overhead with the introduction of P^{2}DAP.

2) Overhead on the DMV: We next examine the number of packets sent to the DMV when an RSB detects suspicious activities and reports to the DMV. This metric is indicative of the communication overhead over the backhaul network connecting the RSB and the DMV. Moreover, the number of packets forwarded by the RSB dictates the computation overhead of the DMV, since the latter must process each of these packets to detect/confirm a Sybil attack.

- *Overhead on the DMV: C-P^{2}DAP*
  We first show the results of the C-P^{2}DAP scheme in Figure 7 and Figure 8. We observe an increase in the number of transmitted packets when the number of coarse-grained hash values increases in Figure 7. This result is because: for a suspected event with a given number of pseudonyms, when the number of coarse-grained hash values increases, the number of packets used to report the event increases. On the other hand, from figure 8, we see the number of pseudonyms from an RSB slightly decreases for increasing number of coarse-grained hash values. While this result seems to contradict the results shown in Figure 7, it can be easily explained as follows: in C-P^{2}DAP, a larger number of coarse-grained hash values result in a smaller number of false alarms.
  From both Figure 7 and Figure 8, we see that the communication overhead of the DMV is very large, thus causing a huge computation overhead to the DMV as well. Therefore, we conclude that, though being able to detect all the malicious behaviors, the C-P^{2}DAP is not scalable to large number of vehicles.
- *Overhead on the DMV: E-P^{2}DAP*
  We next show the relationship between the number of packets from an RSB and the number of benign vehicles in E-P^{2}DAP scheme in Figure 9. In the implementation of this scheme, we split each reported event to several packets such that each RSB report packet contains at most 20 pseudonyms. Therefore, it is not necessary to check the exact number of pseudonyms sent from an RSB, because it can be easily estimated from the number of packets sent from the RSB. By comparing the results in Figure 9 and Figure 7, we find that the packets received by the DMV is much less when using E-P^{2}DAP, which means the E-P^{2}DAP can efficiently distribute the job of detecting Sybil attack the RSBs. Moreover, from Figure 9, we observe that the communication overhead of the DMV almost remains at the same level when the number of benign vehicles increases. We conclude from these observations that the E-P^{2}DAP is scalable to large number of benign vehicles.
- *Overhead on the DMV: T-P^{2}DAP*
  We then examine the DMV overhead in T-P^{2}DAP. Similar to E-P^{2}DAP, each packet from RSBs to the DMV contains an event and a maximum of 20 signing pseudonyms. If an event is signed with more than 20 pseudonyms, the RSBs will split the report into several packets. Figure 10 shows the communication overhead of the DMV in T-P^{2}DAP scheme. We observe an increase in communication overhead when the value of τ increases or the

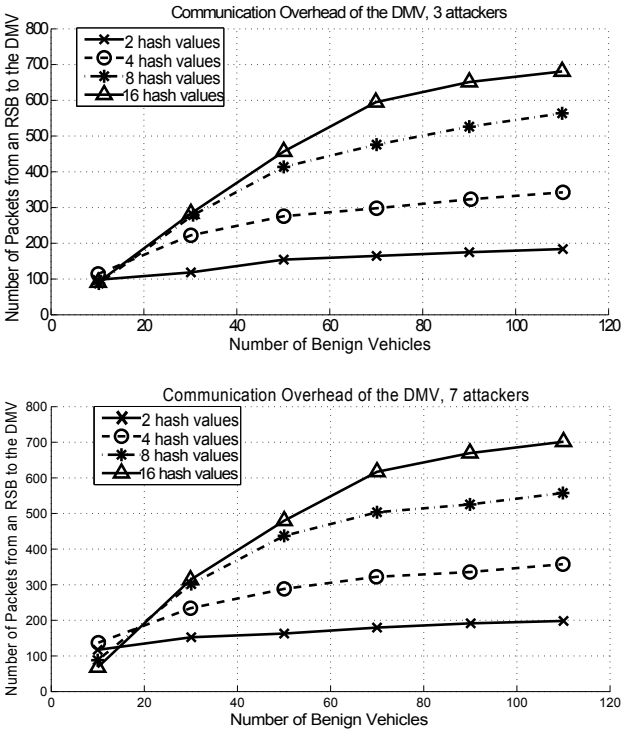


Fig. 7. Number of packets sent from an RSB to the DMV: C-P^{2}DAP

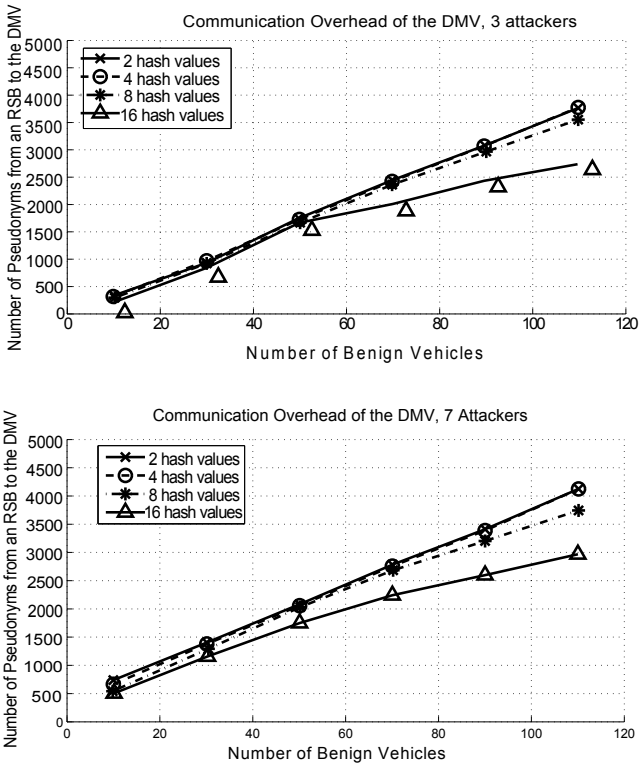


Fig. 8. Number of pseudonyms from an RSB to the DMV: C-P^{2}DAP

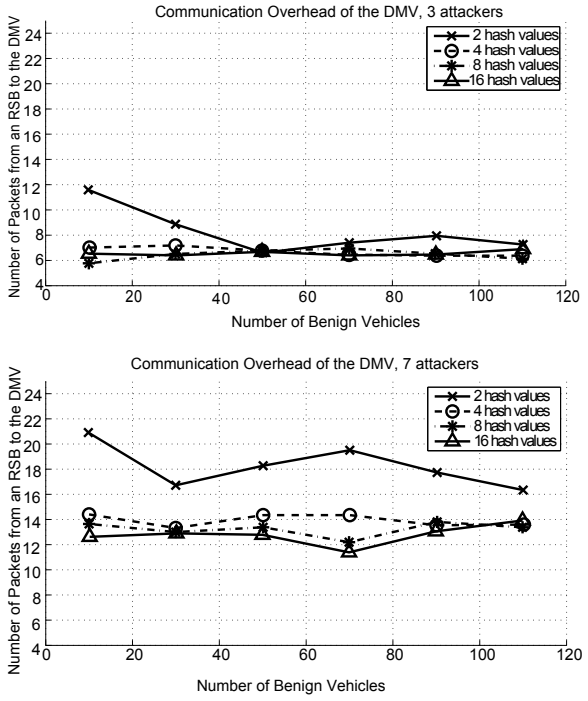


Fig. 9. Number of packets from an RSB to the DMV: E-P^{2}DAP

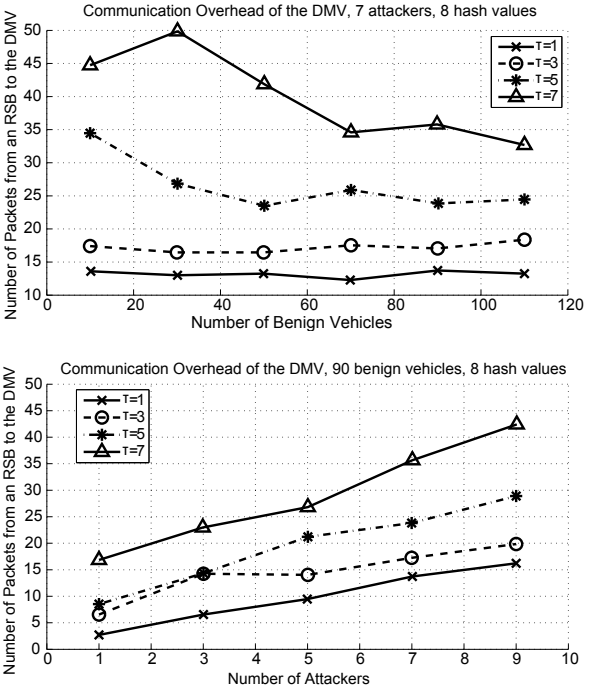


Fig. 10. Number of packets from an RSB to the DMV: T-P^{2}DAP

number of attackers increases. However, the overhead is still much lower than C-P^{2}DAP.

- *Discussion on the Three Schemes*
  As shown above, the C-P^{2}DAP is costly. Comparing to the C-P^{2}DAP, the E-P^{2}DAP has a significant decrease of the communication overhead of the DMV. On the other hand, as a trade-off between C-P^{2}DAP and E-P^{2}DAP, the communication overhead of T-P^{2}DAP is between them, and is adaptive. One interesting observation by comparing

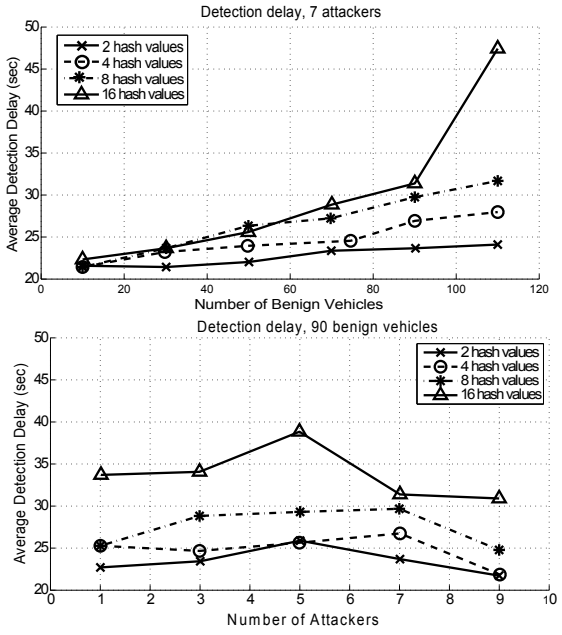


Fig. 11. Detection latency: C-P^{2}DAP

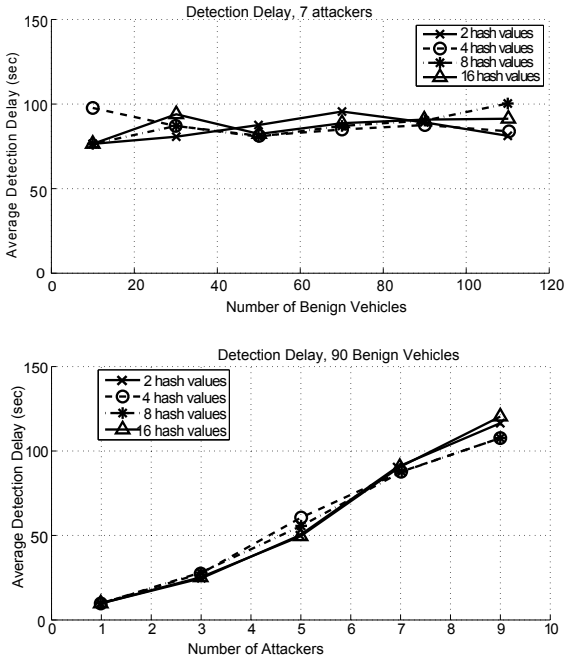


Fig. 12. Detection latency: E-P^{2}DAP

Figure 7 and Figure 10 is that even when τ is only one less than the number of coarse-grained hash values, the communication overhead of T-P^{2}DAP is much less than the C-P^{2}DAP.

E. Simulation Results: Latency for Detecting Malicious Vehicles

In our simulation, the latency Δt for detecting an attacker is defined as $t_{detect} - t_{attack}$, where t_{detect} is the time when the attacker is detected by the DMV, and t_{attack} is the time when the attacker first attacks.

1) C-P^{2}DAP: The C-P^{2}DAP guarantees that every Sybil attack can be detected, therefore Δt is expected to be the shortest. As discussed in Section III, an RSB makes one detection for suspected actions/events at each time interval. Therefore, the earliest time that an attack being caught is in the next time interval of that attack, and Δt is expected to be the length of the time interval.

In Figure 11, we show Δt for the C-P^{2}DAP. We observe that Δt increases for an increasing number of vehicles or an increasing number of Γ (coarse-grained hash values), and Δt is obviously greater than the length of the time interval when the number of vehicles is greater than 90. All these differences are due to a same reason – Recall that in C-P^{2}DAP, a large number of benign vehicles will cause huge communication overhead on the DMV. With a limited bandwidth between the RSB and the DMV, such overhead may cause delay for RSB report (Note that in the simulation, the transmission rate between the RSB and the DMV is 3 pkts/sec), which explains the increase of Δt. Therefore, we conclude that the C-P^{2}DAP scheme, although theoretically guarantees to detect every Sybil attack, may fail or have a large latency on a highly congested road. In real life, we can have much more bandwidth between an RSB and the DMV to solve this issue. However, the result shows a constraint of C-P^{2}DAP – it requires many computation and communication resources to guarantee the successful detections and short detection latency.

2) E-P^{2}DAP: We next examine Δt of E-P^{2}DAP scheme in Figure 12. We observe that the number of benign vehicles has little impact on Δt. This is because probability that all the vehicles reporting an event have the same value of Γ is small. On the other hand, we observe an increasing value of Δt when the number of attackers increases, which can be explained as follows.

Given semi-colluding attackers, the E-P^{2}DAP scheme cannot detect colluding attackers. However, once an attacker reports an event by itself, it will be detected and then revoked. With more attackers being detected, the probability of attackers' collusion decreases, and the remaining attackers are more likely to be detected. In this process, E-P^{2}DAP detects attackers one after another. When there are more attackers, it takes more time for E-P^{2}DAP to observe events signed by only one attacker, thus prolonging Δt.

3) T-P^{2}DAP: The detection delay in T-P^{2}DAP is shown in Figure 13. From the simulation results, we see that by increasing the value of τ, Δt is decreased to around 20s, which is the length of time interval in the simulation. Such results mean that the semi-collusion is resolved by T-P^{2}DAP. Besides, we observe that latency introduced by the communication overhead is not significant. Without this latency, the value of Δt of T-P^{2}DAP is even better than that of C-P^{2}DAP.

4) Discussion of the Above Three Schemes: When designing the above three schemes, according to the resilience to collusion for them, we would expect Δt to be:

$$\Delta t_{\text{C-P}^2\text{DAP}} < \Delta t_{\text{T-P}^2\text{DAP}} < \Delta t_{\text{E-P}^2\text{DAP}}$$

As shown in Figure 12, E-P^{2}DAP has the highest Δt as expected. On the other hand, by comparing Figures 11 and 13,

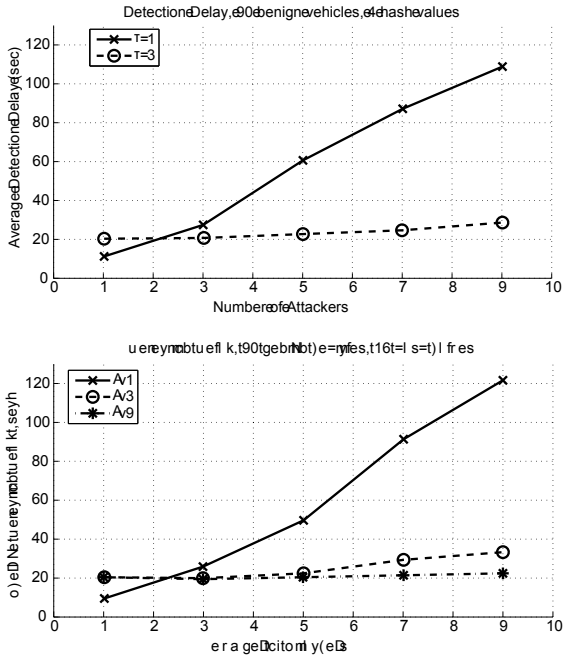


Fig. 13. Detection latency, T-P^{2}DAP, 90 benign vehicles

we observe that due to its light communication overhead, T-P^{2}DAP could win over C-P^{2}DAP in terms of detection latency in the scenario with constrained communication resources.

F. Scenarios with Sparser Vehicle Distribution

In the subsections V-D and V-E, we simulated a road of length 2,000m and vehicles with a radio range of 200m. In such a scenario, each RSB is expected to hear almost all the event reports when the number of vehicles on the road is over 50. The above scenario can simulate the case where the vehicles are highly congested. While in some other cases, vehicles are more sparsely distributed, and not every packet from every vehicle can be captured by the RSB. To simulate such a case, we create a different scenario, in which vehicles go back and forth on a road with a length of 20,000m. Also, the communication radius is set to 50m instead of 200m. Thus, RSBs have less opportunity to overhear vehicles, and the traffic volume near an RSB fluctuates more. We compare the performance of τ-P^{2}DAP and T-P^{2}DAP under this scenario. Considering that both C-P^{2}DAP and E-P^{2}DAP are special cases of T-P^{2}DAP, we do not individually analyze them in this subsection.

1) Communication Overhead of the RSB: We first show the communication overhead of the RSB such that we can compare them to the overhead of the DMV later. As Figure 14 shows, the overhead increases when the number of attackers or the number of benign vehicles grows. This result is similar to the scenario with densely-distributed vehicles, and is an expected behavior.

2) Communication Overhead of the DMV: In Figure 15, we show the communication overhead of the DMV for T-P^{2}DAP and τ-P^{2}DAP. In T-P^{2}DAP, we use 4-bit coarse-grained hash values; while in τ-P^{2}DAP, we use both 3-bit and 4-bit coarse-grained hash values. From the result of T-P^{2}DAP, we observe slight fluctuations in the communication overhead when τ

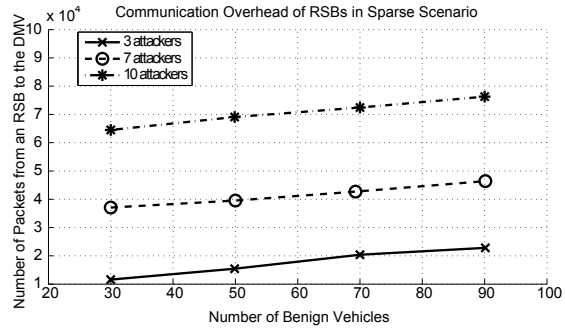
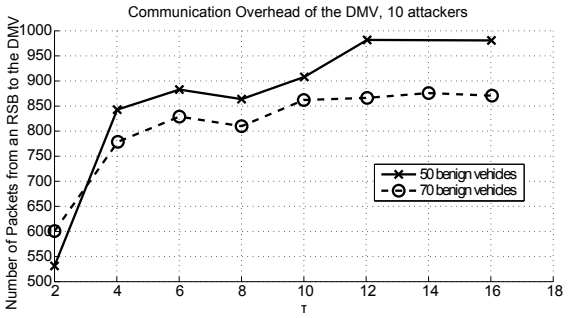
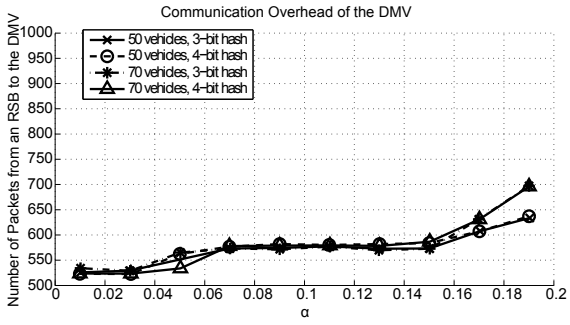


Fig. 14. Communication Overhead of an RSB in Sparse-Vehicle scenario.



(a) T-P^{2}DAP



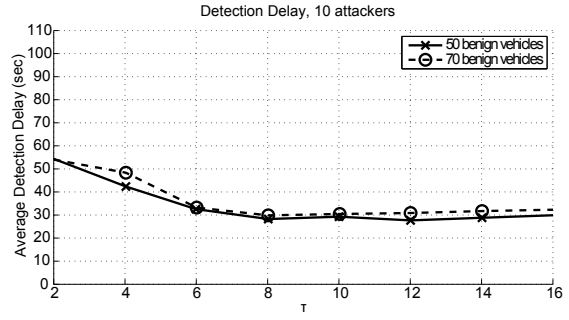
(b) τ-P^{2}DAP

Fig. 15. Communication overhead of the DMV, 10 malicious vehicles.

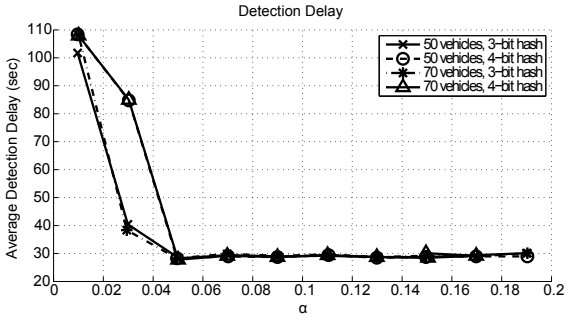
increases. This fluctuation happens because an increasing value of τ can have two opposite results – on one hand, when τ grows, the number of packets forwarded to the DMV grows as well, thus increasing the overhead; on the other hand, a larger value of τ also causes a shorter detection latency, which results in a smaller overall communication overhead. From Figure 15(b), we see that when $\alpha < 0.15$, the communication overhead of τ-P^{2}DAP remains at the same level of the T-P^{2}DAP with $\tau = 2$. When $\alpha > 0.15$, the communication of τ-P^{2}DAP increases dramatically. The results show that when $\alpha \leq 0.15$, we can obtain an acceptable communication overhead.

3) Detection Latency: We next check the value of Δt of the above two schemes. From Figure 16, we observe that Δt of τ-P^{2}DAP quickly drops to 30 seconds when $\alpha > 0.05$, and then maintains at a constant level. On the other hand, T-P^{2}DAP can achieve the similar Δt only when $\tau \geq 8$.

When combined the conclusion from the communication overhead, we conclude that when $0.05 < \alpha < 0.15$, τ-P^{2}DAP



(a) T-P^{2}DAP



(b) τ-P^{2}DAP

Fig. 16. Average detection latency of 10 malicious vehicles.

achieves a good tradeoff between overhead and latency, i.e., Δt reaches 30 seconds at the cost of around 580 packets' communication overhead on the DMV. Such a trade-off cannot be achieved by using a fixed value of τ in T-P^{2}DAP, with the following reason. In T-P^{2}DAP, to achieve the same level of Δt, we require $\tau \geq 8$; while to achieve the same level of communication overhead, we require $\tau \leq 2$. The two requirements of the value of τ cannot be satisfied at the same time. Also, note the actual percentages of attackers are 12.5% (for 70 benign vehicles and 10 attackers) and 16.7% (for 50 benign vehicles and 10 attackers). From these observations, these proportions of attackers among all the vehicles are in the "best" range of α. Therefore, we conclude that, once a proper estimation of α can be made, τ-P^{2}DAP can achieve a satisfying trade-off between communication overhead and detection latency.

VI. Conclusion and Future Work

In this paper, we propose a new method to detect Sybil attacks in VANET. The proposed method distributes the computation workload from the DMV to RSBs while releasing only a limited amount of information by using hash collusions. We also discussed some improvements on our scheme. Based on simulation results presented, we prove that the idea of distributing DMV workload to RSBs with limited information released is applicable in other VANET security and privacy applications.

One interesting future work is to develop a machine-learning algorithm to predict the ratio and activities of malicious vehicles. With a good estimation of the ratio of attackers, P^{2}DAP is expected to efficiently catch attackers with a small overhead and delay. Besides, the DMV can be involved for a centralized management of resources during the detection. Furthermore, the DMV can be distributed to different areas such as regional DMV, which matches the case in real life, and forms a more powerful structure.

Other future work includes developing a more efficient method for partial pseudonym distribution. Moreover, we expect the ideas of distributing the DMV's duty to multiple RSBs for more applications than Sybil detection. Also, in our future experiments, real devices and 802.11p protocols are planned to be used.

Acknowledgment

Ning's work is supported by the National Science Foundation (NSF) under grants CAREER-0447761 and CNS-0721424.

Roy Choudhury's work is supported partially by the NSF under the CAREER award CNS-0747206.

References

[1] F. A. I. W. on Vehicular Ad Hoc Networks (VANET), "Fleetnet: Communication platform for vehicular ad hoc networks," in *Zukunftsforum Mobiles Internet 2010*, October 2004.

[2] T. Kosch and M. Strassberger, "The role of new wireless technologies in automotive telematics and active safety," in *8th Symposium Mobile Communications in Transportation*, 2004.

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

[4] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "Caravan: Providing location privacy for vanet," in *Embedded Security in Cars (ESCAR) Workshop*, 2005.

[5] J. Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks," in *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, 2005.

[6] G. Calandriello, P. Papadimitratos, A. Lioy, and J.-P. Hubaux, "Efficient and robust pseudonymous authentication in vanet," in *ACM International Workshop on VehiculAr Inter-NETworking (VANET)*, 2007.

[7] S. Rass, S. Fuchs, M. Schaffer, and K. Kyamakya, "How to protect privacy in floating car data systems," in *ACM International Workshop on VehiculAr Inter-NETworking (VANET)*, 2008.

[8] B. Parno and A. Perrig, "Challenges in securing vehicular networks," *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, 2005.

[9] A. Studer, E. Shi, F. Bai, and A. Perrig, "Tacking together efficient authentication, revocation, and privacy in vanets," in *Conference on SEnsor, Mesh and Ad Hoc COmmunications and Networks (SECON)*, 2009.

[10] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in vanets," in *ACM International Workshop on VehiculAr Inter-NETworking (VANET)*, October 2004.

[11] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *SASN*, Nov 2005.

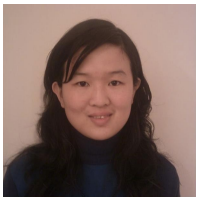
[12] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, and J. pierre Hubaux, "Certificate revocation in vehicular networks," Laboratory for computer Communications and Applications (LCA), School of Computer and Communication Sciences, EPFL, Switzerland, Tech. Rep. LCA-Report-2006-006, 2006.

[13] W. Suh, H. Yun, K. S. Chon, and C. H. Park, "Forecasting hourly traffic volume of airport access road: Case study of incheon international airport," 2005 Annual Transportation Research Boards' (TRB) Meeeting, 2005. [Online]. Available: http://www.leighfisher.com/trb/575-2-suh.pdf

[14] N. Y. S. D. of Transportation, "List of state routes in columbia county," website. [Online]. Available: https://www.nysdot.gov/

[15] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal on Uncertainty,. Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[16] P. Flajolet, D. Gardy, and L. Thimonier, "Birthday paradox, coupon collectors, caching algorithms and self-organized search," *Discrete Applied Mathematics*, vol. 39, pp. 207 – 229, 1992.



Tong Zhou is a software engineer in Spirent Communications, and a part-time PhD student of ECE department at Duke University. She received her bachelor's degree from University of Science and Technology of China, and her master's degree from University of Massachusetts, Dartmouth in 2003. Her research interests are in mobile networks and network security. She has been a PhD student at Duke since 2004.

Romit Roy Choudhury is an Assistant Professor of ECE and CS at Duke University. He joined Duke in Fall 2006, after completing his PhD from UIUC. His research interests are in wireless protocol design mainly at the PHY/MAC layer, and in distributed mobile computing at the application layer. He received the NSF CAREER Award in January 2008. Visit Romit's Systems Networking Research Group (SyNRG), at http://synrg.ee.duke.edu

Peng Ning is a Professor of Computer Science at NC State University, located in Raleigh, NC, USA, where he also serves as the Technical Director for Secure Open Systems Initiative (SOSI) in College of Engineering at NC State University. He joined NC State University in August 2001 after he graduated from George Mason University with a PhD degree in Information Technology. Peng Ning received a BS degree in Information Science and an ME degree in Communication and Electronic System in 1994 and 1997, respectively, both from University of Science and Technology of China.

Peng Ning's research interests are mainly in computer and network security. He is a recipient of NSF CAREER award. His research has been supported by the National Science Foundation (NSF), the Army Research Office (ARO), the Advanced Research and Development Activity (ARDA), IBM Open Collaboration Research (OCR) program, SRI International, and the NCSU/Duke Center for Advanced Computing and Communication (CACC). He was elected the Secretary/Treasurer of the ACM Special Interest Group on Security, Auditing and Control (SIGSAC) in 2009. He served/or is serving on the editorial boards of ACM Transactions on Sensor Networks, Journal of Computer Security, Ad-Hoc Networks, Ad-Hoc & Sensor Networks: an International Journal, International Journal of Security and Networks, and IET Proceedings Information Security. Peng Ning served as the Program Chairs or Co-Chairs of ESORICS 09, ACM SASN 05 and ICICS 06, the General Chair of ACM CCS 07 and CCS 08, and Program Vice Chair for ICDCS 09 & 10Security and Privacy Track. He is a Steering Committee member of ACM CCS and a founding Steering Committee member of ACM WiSec. He has served on the organizing committees or program committees for over fifty technical conferences or workshops related to computer and network security. Peng Ning is a senior member of the ACM, the ACM SIGSAC, and a member of the IEEE and the IEEE Computer Society.

Krishnendu Chakrabarty received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively. He is now Professor of Electrical and Computer Engineering at Duke University. He is also a member of the Chair Professor Group (honorary position) in Software Theory at the School of Software, Tsinghua University, Beijing, China. Prof. Chakrabarty is a recipient of the National Science Foundation Early Faculty (CAREER) award, the Office of Naval Research Young Investigator award, the Humboldt Research Fellowship from the Alexander von Humboldt Foundation, Germany, and four best paper awards at IEEE conferences. His current research projects include: testing and design-for-testability of integrated circuits; digital microfluidics and biochips, circuits and systems based on DNA self-assembly, and wireless sensor networks. He has authored nine books on these topics, published over 350 papers in journals and refereed conference proceedings, and given over 140 invited, keynote, and plenary talks. Prof. Chakrabarty is a Fellow of IEEE, a Golden Core Member of the IEEE Computer Society, and a Distinguished Engineer of ACM. He was a 2009 Invitational Fellow of the Japan Society for the Promotion of Science (JSPS). He is a recipient of the 2008 Duke University Graduate School Deans Award for excellence in mentoring, and the 2010 Capers and Marion McDonald Award for Excellence in Mentoring and Advising, Pratt School of Engineering, Duke University. He served as a Distinguished Visitor of the IEEE Computer Society during 2005-2007, and as a Distinguished Lecturer of the IEEE Circuits and Systems Society during 2006-2007. Currently he serves as an ACM Distinguished Speaker, as well as a Distinguished Visitor of the IEEE Computer Society for 2010-2012. He is the Editor-in-Chief for IEEE Design & Test of Computers and for ACM Journal on Emerging Technologies in Computing Systems. Prof. Chakrabarty is also an Associate Editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Circuits and Systems II, and IEEE Transactions on Biomedical Circuits and Systems. He serves as an Editor of the Journal of Electronic Testing: Theory and Applications (JETTA).