Sybil Attack Resilient Traffic Networks: A Physics-Based Trust Propagation Approach

Yasser Shoukry* Shaunak Mishra Zutian Luo** Suhas Diggavi**

*Department of Electrical and Computer Engineering, University of Maryland, College Park, MD

**Department of Electrical and Computer Engineering, University of California, Los Angeles, CA

yshoukry@ece.umd.edu, {shaunakmishra,zulo,suhasdiggavi}@ucla.edu

***Abstract*—We study a crowdsourcing aided road traffic estimation setup, where a fraction of users (vehicles) are malicious, and report wrong sensory information, or even worse, report the presence of Sybil (ghost) vehicles that do not physically exist. The motivation for such attacks lies in the possibility of creating a "virtual" congestion that can influence routing algorithms, leading to "actual" congestion and chaos. We propose a Sybil attack-resilient traffic estimation and routing algorithm that is resilient against such attacks. In particular, our algorithm leverages noisy information from legacy sensing infrastructure, along with the dynamics and proximity graph of vehicles inferred from crowdsourced data. Furthermore, the scalability of our algorithm is based on efficient Boolean Satisfiability (SAT) solvers. We validated our algorithm using real traffic data from the Italian city of Bologna. Our algorithm led to a significant reduction in average travel time in the presence of Sybil attacks, including cases where the travel time was reduced from about an hour to a few minutes.**

***Keywords*-Secure Smart transportation systems; Sybil attacks; resilient routing;**

I. Introduction

Smart transportation systems hold the promise of radical changes in our daily life. In such systems, sensory information is being collected on an unprecedented scale from vehicles as well as legacy infrastructure (*e.g.*, loop sensors). The data collection mechanism itself has undergone revolutionary changes. Mobile apps like Waze [1] have enabled millions of users to report real-time traffic information; such crowdsourced information is then used to influence routing recommendations. Government agencies, which own the legacy sensing infrastructure, are also actively exploring ways to leverage the data collected by apps like Waze [2] for improving their services. Along with such revolutionary changes in data collection, the sensing capabilities of vehicles have also dramatically increased over time. According to market reports [3], the semi-autonomous vehicles' market was estimated to be 3.17 million units in 2016 and is projected to reach 7.84 million units by 2021. Semi-autonomous and autonomous vehicles can easily infer the position and velocity of nearby vehicles (*e.g.*, through LIDAR and computer vision based methods). Hence, in a crowdsourcing setup, users can not only report their current traffic situation, but they also have the resources to report information regarding their neighbors on the road even without direct vehicle-to-vehicle communication. Given the seriousness of traffic congestion in major cities around the world, it is of significant interest to develop applications leveraging such superior sensing capabilities of vehicles as well as the power of crowdsourcing backed mobile apps like Waze. A natural direction, towards building such applications, is to try to have better real-time estimates of road traffic conditions and decide routing recommendations accordingly. To do so in a secure manner (*i.e.*, resilient to dishonest and fake users) is the primary motivation of this paper.

Crowdsourced data collection methods and subsequent estimation algorithms leave open the possibility of new methods for attacking transportation networks. For example, in 2014, the vulnerability of Waze app to fake accounts was linked to fake traffic jams [4]. In fact, even a very small number of fake cars can lead to severe traffic jams [5], [6]. Such Sybil attacks, where an attacker creates fictitious cars (users), and then reports false information are not unheard of; they are quite prevalent in peer-to-peer networks [7] and online social networks [8], [9]. Existing approaches for mitigating such Sybil attacks include: (i) graph-based methods [6], [8], [10]–[12], and (ii) authentication-based on wireless signals [13]–[16]. Graph-based methods primarily rely on the assumption that fake users have difficulty getting connected to real users, and tend to form closed clusters within themselves (*i.e.*, the structure of *proximity* graph). Such assumption is violated in traffic systems since a real car can be the one that launches the Sybil attack in which case the Sybil cars can report precisely the dishonest car. Since this dishonest car physically exists and hence will be reported by the other real cars which themselves are reported by other real cars, the assumption that Sybil cars form closed clusters within themselves no longer holds. Therefore, algorithms based on such heuristic are not guaranteed to detect the existence of Sybil attacks. Authentication methods based on wireless signals are closely tied to the physical layer properties of wireless signals. For example, [16] created spatial fingerprints based on the multipath propagation of wireless signals; this suffers from free space environments (*i.e.*, the absence of multipath).

Directly applying existing Sybil attack mitigation methods (as mentioned above) to the scenario we consider in this paper (*i.e.*, traffic estimation and routing based on crowdsourced data) will miss a significant aspect: a transportation system is necessarily a dynamical system whose state evo-

This work was supported in part by NSF award #1740047.

0-7695-6378-3/18/$31.00 ©2018 IEEE
DOI 10.1109/ICCPS.2018.00013

lution, *i.e.*, the manner in which the traffic density at a given location changes over time, is governed by physics. In fact, it is well established in the area of cyber-physical systems (CPS) security that, the knowledge of system dynamics can be leveraged to fundamentally secure CPSs against malicious attacks [17]–[23] and provide an additional layer of security on top of existing network security approaches. Dynamics-based CPS security approaches like the ones mentioned above are certainly relevant to the crowdsourcing based traffic estimation setup; vehicles/users can be treated as (noisy) sensors in a dynamical system (transportation networks) where the goal is to estimate the underlying state (traffic conditions). However, there are multiple shortcomings in the existing CPS security literature in the context of Sybil attacks in a transportation network: (i) while many papers in the literature studied the problem of false data injection, Sybil attacks where dishonest sensors are allowed to introduce additional fake sensors did not have much attention, (ii) previous results in the literature assume sparse attacks (which is violated if a single dishonest sensor is allowed to introduce many fake sensors), and (iii) previous work has not leveraged the proximity graph formed by sensors (vehicles). In the context of such shortcomings, our contributions in this paper can be summarized as follows:

- We develop an a traffic state estimation approach for inferring a reliable subset of cars in a road segment. Our approach takes into account the information from noisy legacy infrastructure (loop sensors), the proximity graph of vehicles participating in crowdsourcing, vehicle dynamics as well as benign noise in the crowdsourced data.
- We introduce a physics-based trust propagation mechanism to further strengthen our inference of reliable cars.
- We develop theoretical guarantees on the estimation error in the presence of attacks, as well as demonstrate significant performance improvements on a real traffic dataset obtained from the Italian city of Bologna [24]. In particular, we show that the travel time which could be about an hour in the presence of Sybil attacks, can be reduced to a few minutes using our algorithm.

II. Problem Setup and Threat Model

As shown in Figure 1, we consider a traffic network that is equipped with legacy infrastructure (e.g., inductive loop sensors or cameras) that provides real-time estimates of the traffic condition. Due to cost and maintenance constraints, these legacy sensors can not instrument the whole traffic network, and hence they are placed many kilometers apart from each other. Moreover, due to aging, several of these sensors are not reliable (report very noisy measurements) and cannot be exclusively used to provide accurate estimates of traffic conditions to regulate the traffic network and achieve its highest performance.

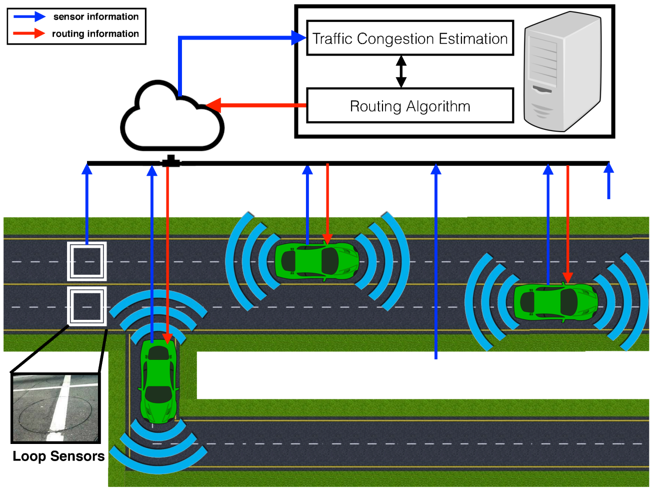

Figure 1. A cartoon demonstrating the problem setup under consideration. Smart cars use their built-in sensors to measure their position, velocity and the velocity of the nearby vehicles. These sensory data along with data collected from legacy infrastructure (e.g., cameras and loop sensors) are then sent to a cloud-based server which estimates the traffic conditions. The estimated traffic conditions are then used to generate routing trajectories for each car to reduce the average travel time and balance the congestion through the traffic system.

In an attempt to compute better traffic condition estimates, the traffic network depends on real-time crowdsourced information collected from all the cars. In particular, each car is assumed to use its sensors to report its own velocity (e.g., on-board IMUs and odometers measurements), its position (e.g., GPS measurements), along with the velocities of all nearby cars (measured using radars). We denote by $car_i(t) = (p_i(t), v_i(t), v_i^{j_1}(t), \ldots, v_i^{j_{n_i}}(t))$ the set of measurements collected by the ith car and communicated back to the infrastructure where $p_i(t)$ is the ith car position, $v_i(t)$ is the ith car velocity, and $v_i^j(t)$ is the velocity of the jth car as measured by the ith car radars with $j \in \{j_1, \ldots, j_{n_i}\}$ and n_i is the number of cars in the range of the ith car radars.

The information collected from the infrastructure along with those crowdsourced from all vehicles is then used to generate an estimate for the state of the traffic network. In particular, it is a common practice to split the traffic network into sectors, each of a typical length of $5km$ [25] and compute an estimate of the traffic state (average velocity) for each of these sectors. We denote by K the number of sectors in the traffic system and by n_k the number of cars in the kth sector. Finally, the smart traffic system uses the estimated traffic condition (average velocity for each sector) to compute optimal routes for each car that increase the performance of the whole traffic system.

A. Threat Model

As discussed in the previous subsection, for the smart traffic system to maximize its performance, traffic information from individual vehicles is needed. This opens the question of what happens when the crowdsourced information, reported by different vehicles, is corrupted in an orchestrated fashion. Such corrupted data can be either due to a physical attack on sensors [26], [27], an attack on the vehicle software, or due to an attack on the communication channel between the vehicles and the infrastructure. Regardless of how the attack is taking place, we consider the following two threat models:

1) **False data injection attack:** In such attack, a car that physically exists on the road is reporting maliciously corrupted information (wrong position, speed, and/or speed of nearby vehicles). We refer to these cars as dishonest cars.
2) **Sybil attack:** In this attack, a car which physically exists on the road reports the presence of nearby cars that do not physically exist (ghost cars). Those ghost cars may also report the presence of more nearby ghost cars which in turn report the presence of more ghost cars until a chain of ghost cars are created. Creating such a chain of ghost cars will allow the attacker to gain a disproportionately large influence on the crowdsourced data reported to the traffic infrastructure and hence affect its routing decisions. Following the same terminology of attacks on reputation systems, we refer to these ghost cars as Sybil cars [7].

To understand the consequences of an orchestrated Sybil attack, we consider the situation reported in Figure 2 in which a malicious car (red) is traveling on a highway with light traffic conditions (moderate density/congestion and high average velocity). This malicious car launches a Sybil attack in which it faithfully reports its position but reports a lower traveling speed along with the presence of Sybil cars (yellow) in its surrounding. Next, the attacker takes the identity of one of those Sybil cars to report the presence of more Sybil cars creating a long chain of Sybil cars. All these Sybil cars are reporting a consistent low traveling speed. As a consequence, the traffic controller will indicate an erroneous heavy traffic condition (high density/congestion and low traveling speed) in this particular sector. This incorrect heavy traffic conditions will force the traffic controller to re-route honest cars (green) into the sideways in an attempt to reduce the congestion of the highway and hence maximize the performance of the traffic system. Finally, this diversion will create actual heavy traffic on the side road leading to worse traffic in the whole system.

The primary goal of this paper is to design resilient traffic estimation algorithms that are capable of identifying the existence of such attacks and isolate the set of maliciously reported information. Similarly to the previous work on secure state estimation [17], [18], [21], [28], we assume no prior knowledge of the temporal, magnitude, or stochastic properties of the attack. However, differently from the previous work on secure state estimation where the maximum number of malicious information is known a priori, we assume no such prior information. We only assume that the attacker has no access to the data collected by the legacy sensors (secure sensors). That is, while the attacker may compute an estimate of these measurements, he does not have direct access to them. We argue in this paper that these, typically low-quality, noisy, and spatially sparse sensor measurements, are enough to detect and isolate such attacks.

Finally, we assume that the cloud-based traffic controller is honest in the sense that it computes the optimal routing information based on the received sensor information.

B. A Note on Confidentiality Attacks

According to the threat model discussed in the previous section, we consider only "active" attacks in which the attacker corrupts the integrity of the sensory information exchanged crowdsourced and exchanged within the system. Another related concern is the privacy or the confidentiality of this sensor information which are shared by the users with the traffic infrastructure. In this paper, and to solicit participation from users, we assume that all sensor and routing information exchanged between the individual cars and the traffic controller are encrypted before sent on the network (to prevent eavesdropping in the network). This can be achieved using a public key encryption scheme. Moreover, we assume that all computations at the traffic controller are executed using a trusted computation platform (e.g., secure enclaves in Intel SGX) which prevents other software running on the traffic controller from accessing the shared sensor information. Such technologies are adopted and featured by many cloud servers (e.g., Microsoft Azure [29]) to provide secure cloud computing. Therefore, within this paper, will focus only on active integrity attacks as discussed in the previous subsection.

III. Attack Detection Using Physics Based Trust Propagation

Trust propagation is a cyber-security principle by which new trust relationships can be derived from pre-existing trust relationship. That is, in each system, we start by searching for a subsystem (a hardware component, software, or authority) that can always be trusted. Such subsystem is typically named the root-of-trust. Next, we use this root-of-trust along with a set of rules named trust transitivity and trust fusion to "propagate" the trust embodied in the root-of-trust subsystem into a trust in the rest of subsystems [30]. This principle has been extensively used in software and hardware security [31], and secure recommendation in social networks [32]. In this section, we discuss how to use the same principle to build attack-resilient traffic systems. In particular, we identify roots-of-trust in the traffic system along with defining rules for trust propagation across the traffic network.

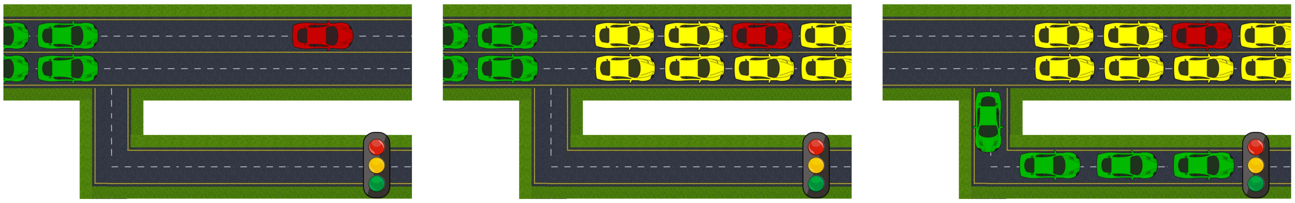

Figure 2. A scenario showing the consequences of a Sybil attack. This scenario consists of a highway and a sideway: (a) A malicious car (red) is traveling over the highway, and benign vehicles (green) are entering the highway while the sideway is congestion-free; (b) The malicious car (red) reports the existence of several Sybil cars (yellow) on the highway which they do not physically exist creating "virtual" congestion on the highway; (c) the smart traffic system re-routes the good cars (green) into the sideway in an attempt to reduce the congestion on the highway which leads to a "real" congestion on the sideway.

A. Legacy Sensors as a Root-of-Trust

As discussed in Section II, only few sectors of the traffic network are equipped with legacy sensors (e.g., induction loop sensors and cameras) which provide estimates of traffic flow and congestion for only those instrumented sectors. Nevertheless, these noisy and spatially sporadic measurements are assumed to be attack-free and hence considered as the first root-of-trust for the traffic network.

Reilly *et al.*in [26] reports a thorough study on the vulnerability of these legacy sensors. It is shown that these sensors have two main vulnerabilities namely (i) physical access to the sensor hardware (copper theft, wire clipping) and (ii) communication links between the sensor and the rest of the infrastructure. We argue that physical access to the sensor hardware (e.g., copper/wires used by the loop sensors) is a complicated process since these sensors are buried within the asphalt. Similarly, for cameras, tampering with cameras held on traffic poles in the highways is similarly a laborious process and digital authentication techniques can be used to detect any replacement of these cameras. Moreover, as noted by [26], attacks on these sensors may not be enough to launch an influential attack. Therefore, given the barriers to physical access and the limitation of influence, we argue that this attack is not attractive for attackers. Similarly, for attacks on communication channel between these sensors and the rest of infrastructure, we argue that, given the current advances in cryptography (see for example [33] and the references within), this communication link can be designed to be resilient against such attacks and hence secure this root-of-trust.

B. Physics-Based Trust Propagation

In order to propagate the trust from the sectors instrumented by legacy sensors to all other sectors, we propose a physics-based trust propagation scheme. In such scheme, we rely on physics-based models that relate the traffic state (cars density and average velocity) at one sector to the traffic states at the adjacent sectors. Such physics-based models have been studied extensively in the transportation literature [34] where several models with increasing complexities have been proposed and verified. In particular, we rely on a second-order validated macroscopic model which describes the dynamic behavior of traffic flow along a freeway sector in terms of aggregate flow variables [25], [35]. For the convenience of computation, this macroscopic model is presented in a space-time discretized form as follows. Assuming that the traffic inflow for the kth traffic sector enters from the $k-1$ sector and the on-ramps inside the kth sector and the outflow of traffic from the kth sector exists to the $k+1$ sector and the off-ramps (if any), the dynamics of the traffic state in the kth sector can be written as:

$$\rho_k(t+1) = \rho_k(t) + \frac{T_s}{\Delta_k \lambda_k}[q_{k-1}(t) - q_k(t) + r_k(t) - s_k(t)] \tag{1}$$

$$\begin{aligned} v_k(t+1) = v_k(t) &+ \frac{T_s}{\tau}[V(\rho_k(t)) - v_k(t)] \\ &+ \frac{T}{\Delta_k} v_k(t)[v_{k-1}(t) - v_k(t)] \\ &- \frac{\nu T_s}{\tau \Delta_k} \frac{\rho_{k+1}(t) - \rho_k(t)}{\rho_k(t) + \kappa} - \frac{\delta T_s}{\Delta_k \lambda_k} \frac{r_k(t) v_k(t)}{\rho_k(t) + \kappa} \\ &+ \zeta_k(t) \end{aligned} \tag{2}$$

where:

- $\rho_k(t)$ is the traffic density in the kth segment at time instant t which is defined as the number of vehicles in that segment normalized by the sector length Δ_k and the number of lanes λ_k,
- $v_k(t)$ is the average speed of all vehicles included in the kth segment,
- $q_k(t)$ is the outgoing traffic flow which is defined as the number of vehicles leaving the kth segment during the time period $[t, t+T_s]$ normalized by the sampling time T_s,
- $r_k(t)$ and $s_k(t)$ are the on-ramp inflow and off-ramp outflow (if a ramp exists within that sector),
- $V(\rho)$ is the stationary speed equation [25], [35],
- $\zeta_k(t)$ is a zero-mean noise that represents any model inconsistencies. Note that equation (1) is not corrupted by noise because it models the conservation of vehicles which holds strictly in all cases.

Assuming unknown and time-varying model parameters and inductive loop sensors are placed several sectors apart, Wang et al. [25], [35] developed an extended Kalman filter which computes an estimate of the traffic density and average velocity for all sectors which are not instrumented by inductive loop sensors [25], [35]. However, the quality of the produced estimate deteriorates in the sectors that are further away from the loop sensors. We denote by $\hat{v}_k^l, \rho_k^l(t)$ (with $k \in \{1, \ldots, K\}$) the estimates that are produced by the extended Kalman filter developed in [25], [35] where the superscript l reflects the fact that these estimates are computed using the information collected from the legacy sensors only. In other words, using the physics-based model (equations (1)-(2)), we can propagate the information collected from the root-of-trust (legacy sensors) to create a noisy estimate for the traffic state for all the sectors in the transportation network which will be used later to detect and mitigate attacks.

C. Resilient Traffic Estimator

The proposed resilient estimator works as follows. First, we use the position information p_i reported by individual cars to associate them with the corresponding traffic sectors. The next step is to enumerate all possible subsets of vehicles whose sensors report consistent information. An example of inconsistent information is when car A indicates the presence of car B in its neighborhood without car B reporting A. This case would occur when:

- A Sybil car (car A) reports the presence of an honest car (car B). Being honest, car B does not participate in the Sybil attack and hence is not aware of the existence of the Sybil car (car A) and won't report car A accordingly.
- A dishonest car (car B), which physically exist on the road, decides not to report the presence of an honest nearby vehicle (car A) in an attempt to disturb the traffic estimation. On the other side, the honest vehicle reports the presence of the dishonest car in its neighborhood faithfully.

From these two cases, we note the following. In the first case, the attacker vehicle is the one providing information about the presence of another car (honest) while in the second instance the honest vehicle is the one that provided the information about the presence of the other vehicle (dishonest). That is, while inconsistencies between reported information can be used to detect the existence of an attack, information inconsistencies cannot be directly used to identify the malicious vehicle. Moreover, and as discussed in Section II-A, we do not assume the prior knowledge of the maximum number of Sybil and dishonest cars. This, in turn, eliminates the possibility of using heuristics based on the size of vehicles reporting consistent information. Finally, we note that the same vehicle may belong to more than one subset of cars which report consistent information. We denote by $\mathbb{S}_k$ the set whose elements are all the subsets of vehicles reporting consistent information inside the kth sector. Details of computing $\mathbb{S}_k$ are given in the subsequent subsections.

Once the set $\mathbb{S}_k$ is computed, we use the trusted information collected by the legacy sensors and propagated through the entire traffic system using the physics-based model (equations (1)-(2)) to sanitize the data received from each subset of cars in $\mathbb{S}_k$. In particular, for each subset of vehicles with consistent information $S_k \in \mathbb{S}_k$, we compute the average discrepancy (over a window of length N) between their reported velocities and the velocity estimated by the legacy sensors $\hat{v}_k^l$ as:

$$\hat{e}_{S_k}(t) = \frac{1}{N} \sum_{t=t_1}^{t_1+N-1} (\hat{v}_{S_k}(t) - \hat{v}_k^l(t))^2 \qquad \forall S_k \in \mathbb{S}_k$$

where $\hat{v}_{S_k}(t)$ is the average speed among all vehicles indexed by the set S_k, i.e.,

$$\hat{v}_{S_k}(t) = \frac{1}{|S_k|} \sum_{i \in S_k} v_i$$

and $|S_k|$ is the cardinality of the set S_k (which corresponds to the number of cars inside this set). The final step is to choose the subset of vehicles S_k^* which lead to the minimum discrepancy, i.e.,

$$S_k^* = \arg\min_{S_k} \hat{e}_{S_k}$$

The final estimate of the traffic average speed is then computed using only the information provided by the cars indexed by the set S_k^* as follows:

$$\hat{v}_{S_k^*}(t) = \frac{1}{|S_k^*|} \sum_{i \in S_k^*} v_i(t)$$

Traffic density and congestion can also be computed directly from the estimated average velocity and the maximum allowed velocity in the kth sector.

In other words, the noisy estimates that are provided by the legacy sensors and propagated using the physics-based model—which in turns adds more noise and hence reduces its estimation quality—are used only to identify which set of vehicles are reporting honest information. The final estimate of the traffic information is computed using the data published by the honest cars which are assumed to be more accurate. This procedure is summarized in Algorithm 1. It follows from our analysis in Section IV that Algorithm 1 is optimal. In particular, Algorithm 1 can select the subset of vehicles S_k^* whose reported information is the closest to the ground truth regardless the fact that noisy estimates produced by legacy sensors $\hat{v}_k^l(t)$ are used to select the subset S_k^*.

D. Enumerating all consistent vehicles using SAT solver

Enumerating all sets of consistent cars $\mathbb{S}_k$ is a combinatorial search problem in which one needs to take into account different combinations of whether each vehicle is honest/dishonest/Sybil. Therefore, to enumerate all possible

Algorithm 1 Resilient Traffic Estimator

Input: $car_1, \ldots, car_{n_k}, \hat{v}_k^l$ **Output:** $\hat{v}_k(t)$

```
Enumerate all sets of consistent cars
    $\mathbb{S}_k = \textsc{Enumerate-Consistent-Cars}(car_1, .., car_{n_k})$
for Each consistent set of cars $S_k \in \mathbb{S}_k$ do
    Compute the average velocity reported by cars:
        $\hat{v}_{S_k}(t) = \frac{1}{|S_k|} \sum_{i \in S_k} v_i$
    Compute the discrepancy in reported velocity:
        $\hat{e}_{S_k}(t) = \frac{1}{N} \sum_{t=t_1}^{t_1+N-1} (\hat{v}_{S_k}(t) - \hat{v}_k^l(t))^2$
end for
Choose the set with lowest discrepancy:
    $S_k^* = \arg\min_{S} \hat{e}_{S_k}$
Compute the traffic state reported by $S_k^*$:
    $\hat{v}_{S_k^*}(t) = \frac{1}{|S_k^*|} \sum_{i \in S_k^*} v_i(t)$
Return $\hat{v}_{S_k^*}(t)$
```

solutions, we resort to automated reasoning techniques in which the problem is encoded as a set of constraints and a solver is then used to enumerate all possible solutions to these set of constraints. The set $\mathbb{S}_k$ corresponds to all the solutions of these constraints.

We start by defining two Boolean variables for each car in the kth sector. The first Boolean variable $\textsc{isSybil}_i$ is set to zero whenever the ith vehicle is assumed to be physically present and one if the vehicle is assumed to be a Sybil car. The second Boolean variable $\textsc{isHonest}_i$ is set to zero whenever the vehicle is assumed to be dishonest and one otherwise. Indeed, whenever a car is honest, then it is also not Sybil and the following constraint is generated for each vehicle:

$$\textsc{isHonest}_i \Rightarrow \neg\textsc{isSybil}_i \tag{3}$$

The next step is to define a rule that checks whether each pair of cars is reporting consistent information. To that end, we define the Boolean variable $\textsc{velMatching}_{i,j}$ for each pair of cars. This Boolean variable should be set to one if the velocity of the jth car reported by itself is matching the one reported by the ith car radar and vice versa. This constraint can be encoded as:

$$\textsc{velMatching}_{i,j} \Leftrightarrow |v_i - v_j^i| \leq \varepsilon \wedge |v_j - v_i^j| \leq \varepsilon$$

Using these three Boolean variables, we can encode all different scenarios. Note that we generate constraints governing only the scenarios that occur between each pair of cars. However, the solution to all the generated constraints will lead to solutions that consider transitivity between different car agreements. That is, if the reports from car A and B are consistent as well as those from B and C, the solution to this set of constraints will ensure that cars A, B and C are considered consistent together.

The first scenario is when both the two vehicles report the presence of each other. In such scenario, if both cars report consistent velocity information, then there is a possibility that both cars are honest. The fact that two cars are mutually reporting their presence could stem from the case when both cars are Sybil, one car is Sybil while the other one is dishonest, or one of the cars is honest while the second one is dishonest but not Sybil. The following set of constraints capture all these different scenarios and are generated only when the two cars i and j report the presence of each other as a nearby car:

$$\begin{aligned}
&(\textsc{velMatching}_{i,j} \wedge \textsc{isHonest}_i \wedge \textsc{isHonest}_j) \\
&\vee(\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j \wedge \neg\textsc{isSybil}_j) \\
&\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j) \\
&\vee(\textsc{isSybil}_i \wedge \textsc{isSybil}_j)
\end{aligned} \tag{4}$$

Similarly, when only one of the cars i and j report the presence of the other as a nearby car, this rules out the possibility that both the cars are honest. This leaves the possibility that one of the two cars is honest while the other is dishonest/Sybil or that both cars are dishonest/Sybil cars. The following set of constraints capture these different scenarios and are generated only when the ith car report the jth car as a nearby car but not vice versa:

$$\begin{aligned}
&(\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j \wedge \neg\textsc{isSybil}_j) \\
&\vee(\textsc{isHonest}_i \wedge \textsc{isSybil}_j) \\
&\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isSybil}_i \wedge \neg\textsc{isHonest}_j \wedge \neg\textsc{isSybil}_j) \\
&\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isSybil}_i \wedge \textsc{isSybil}_j) \\
&\vee(\textsc{isSybil}_i \wedge \textsc{isSybil}_j)
\end{aligned} \tag{5}$$

Finally, in order to enumerate all possible solutions of the constraints (3)-(5), we use a Boolean satisfiability solver (SAT solver). Invoking the SAT solver with constraints (3)-(5) will result into one assignment of the Boolean variables, $\textsc{isHonest}_i, \textsc{isSybil}_i$, to zeros and ones indicating *one* possible set of consistent and honest cars. This possible set of honest car is then added to the set $\mathbb{S}_k$. Recall that we are interested in enumerating all possible sets of honest cars that satisfy the constraints (3)-(5). Therefore, we need to invoke the SAT solver multiple times until no more possible sets of honest cars can be found. To ensure that the SAT solver will not produce the same assignment multiple times, we need to add the following constraint iteratively:

$$\bigvee_{i \in S_k} \neg\textsc{isHonest}_i \qquad \forall S_k \in \mathbb{S}_k$$

Algorithm 2 Enumerate-Consistent-Cars-SAT-Solver

Input: $car_1, \ldots, car_{n_k}$ **Output:** $\mathbb{S}_k$

```
$\mathbb{S}_k = \emptyset$
Add the following constraints to the SAT solver:
    $\textsc{isHonest}_i \Rightarrow \neg\textsc{isSybil}_i \quad \forall i \in \{1, \ldots, n_k\}$
for $i, j \in \{1, \ldots, n_k\}$ and $i \neq j$ do
    if $|v_i - v_j^i| \leq \varepsilon \wedge |v_j - v_i^j| \leq \varepsilon$ then
        $\textsc{velMatching}_{i,j}$ := True
    end if
    if both $car_i$ and $car_j$ report each other: then
        Add the following constraints to the SAT solver:
            $(\textsc{velMatching}_{i,j} \wedge \textsc{isHonest}_i \wedge \textsc{isHonest}_j)$
            $\vee(\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j \wedge \neg\textsc{isSybil}_j)$
            $\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)$
            $\vee(\textsc{isSybil}_i \wedge \textsc{isSybil}_j)$
    end if
    if $car_i$ reports $car_j$ but not vice versa: then
        Add the following constraints to the SAT solver:
            $(\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j \wedge \neg\textsc{isSybil}_j)$
            $\vee(\textsc{isHonest}_i \wedge \textsc{isSybil}_j)$
            $\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isSybil}_i \wedge \neg\textsc{isHonest}_j$
                $\wedge \neg\textsc{isSybil}_j)$
            $\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isSybil}_i \wedge \textsc{isSybil}_j)$
            $\vee(\textsc{isSybil}_i \wedge \textsc{isSybil}_j)$
    end if
end for
while SAT solver can not find more solutions do
    $(\textsc{isHonest}_i, \textsc{isSybil}_i)$ = SAT-solver()
    $S = \{i \mid \textsc{isHonest}_i = 1\}$
    $\mathbb{S}_k = \mathbb{S}_k \cup S$
    Add the following constraints to the SAT solver:
        $\bigvee_{i \in S} \neg\textsc{isHonest}_i$
end while
Return $\mathbb{S}_k$
```

whenever a new assignment of honest cars S_k is added to the set $\mathbb{S}_k$. The above constraint prevents the SAT solver from generating an assignment that was produced previously. Algorithm 2 summarizes the above discussion.

E. Enumerating all consistent cars using Majority voting

The Boolean encoding in Algorithm 2 uses two Boolean variables for each vehicle in the kth sector. This leads to a search space of size equal to 2^{2n_k} where n_k is the number of cars in the kth sector. To harness this combinatorial growth in the search space, we propose a heuristic that will lead to a search space equal to 2^{n_k} which is significantly smaller than the original search space.

The basic idea is to use only one Boolean variable $\textsc{isHonest}_i$ for each vehicle in the kth sector of the traffic network. This Boolean variable is used to enumerate all the cases of whether the ith vehicle is honest or dishonest/Sybil. Similarly to the previous encoding, we focus only on generating constraints governing the interactions between each pair of cars and let the SAT solver find assignments that satisfy the connectivity of the aggregate reported information. We proceed with case analysis as follows. The first scenario is when both car i and car j report their mutual presence consistently. In such case, either both cars are honest, both the cars are Sybil and mutually report their presence in the same neighborhood, or even worse a dishonest car is launching a Sybil attack and hence reports the presence of another Sybil car in its neighborhood. The last case occurs when an honest vehicle is reporting a dishonest vehicle while the dishonest vehicle maliciously reports back information consistent with the honest one. All these cases can be encoded as:

$$\begin{aligned}
&(\textsc{isHonest}_i \wedge \textsc{isHonest}_j)\\
&\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)\\
&\vee(\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)\\
&\vee(\neg\textsc{isHonest}_i \wedge \textsc{isHonest}_j) \qquad (6)
\end{aligned}$$

The second scenario is when the jth vehicle reports the presence of the ith one but not vice versa. This inconsistency in the reported information eliminates the case when both vehicles are honest and can leave all other possibilities. This can be encoded as:

$$\begin{aligned}
&\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)\\
&\vee(\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)\\
&\vee(\neg\textsc{isHonest}_i \wedge \textsc{isHonest}_j) \qquad (7)
\end{aligned}$$

Note that the constraints generated in (6)- (7), do not take consistency in reported velocities into considerations but only consistency in reporting mutual presence. Therefore, the subsets of cars returned by this encoding may not contain consistent velocity information. The next step is then to use a majority voting over noisy measurements to obtain a robust velocity estimate for each subset of vehicles, i.e., we replace line 3 in Algorithm 1 with:

$$\hat{v}_S(t) = \texttt{NoisyMajorityVoting}_{i \in S}(v_i)$$

Our heuristic is that the majority voting will be able to remove the effect of inconsistencies between reported vehicle information and filters out malicious reports from the subsets dominated by honest vehicles. We refer to this scheme as Majority-voting based scheme and refer to the one in Algorithm 2 as a SAT-based scheme.

Algorithm 3 Enumerate-Consistent-Cars-Majority-Voting
Input: $car_1, \ldots, car_{n_k}$ **Output:** $\mathbb{S}_k$

```
$\mathbb{S}_k = \emptyset$
for $i, j \in \{1, \ldots, n_k\}$ and $i \neq j$ do
    if Car both $car_i$ and $car_j$ report each other: then
        Add the following constraints to the SAT solver:
            $(\textsc{isHonest}_i \wedge \textsc{isHonest}_j)$
            $\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)$
            $\vee(\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)$
            $\vee(\neg\textsc{isHonest}_i \wedge \textsc{isHonest}_j)$
    end if
    if $car_i$ reports $car_j$ but not vice versa: then
        Add the following constraints to the SAT solver:
            $\vee(\neg\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)$
            $\vee(\textsc{isHonest}_i \wedge \neg\textsc{isHonest}_j)$
            $\vee(\neg\textsc{isHonest}_i \wedge \textsc{isHonest}_j)$
    end if
end for
while SAT solver can not find more solutions do
    $\textsc{isHonest}_i = $ SAT-solver()
    $S = \{i \mid \textsc{isHonest}_i = 1\}$
    $\mathbb{S}_k = \mathbb{S}_k \cup S$
    Add the following constraints to the SAT solver:
            $\bigvee_{i \in S} \neg\textsc{isHonest}_i$
end while
Return $\mathbb{S}_k$
```

F. Trusted Cars: Yet Another Root-of-Trust

So far, we based the proposed estimator on the fact that legacy sensors provide a root-of-trust whose trust can be propagated through the entire network to identify malicious behaviors. We note that another possible root-of-trust is the sensory information collected from attack-proof vehicles like police cars for example. That is, one can argue that such vehicles, although corresponding to a small number of cars in the whole system, can be equipped with hardware that is tamper proof and guaranteed to report correct information. This information needs to be propagated through the traffic network to identify malicious behaviors. Thanks to the Boolean encoding discussed in the previous two subsections, we can utilize this additional root-of-trust by assigning the corresponding Boolean variables $\textsc{isHonest}_i$ to True. This, in turn, forces the SAT solver to consider only the subsets of cars which report information consistent with these trusted cars.

IV. Theoretical Guarantee

In this section, we discuss the theoretical guarantees for Algorithm 1 when utilizing the SAT-based scheme. In our analysis, we assume that the velocity estimates computed by the Kalman filter $\hat{v}^l(t)$ from the legacy sensors are equal to the ground truth of the traffic state corrupted by a Gaussian noise $\eta(t)$ with zero mean and unknown variance, i.e.,

$$\hat{v}^l(t) = v(t) + \eta(t)$$

Note that for simplicity of notation, we drop the subscript k in our analysis. In particular, we show that Algorithm 1 is "optimal" and reports back a traffic state estimate whose quality (or error) is the closest to the one reported by the honest (or "attack-free" vehicles). This is captured by the following theorem whose proof can be found in [36].

Theorem 1. *Consider the traffic system under false data injection and Sybil attacks. Let S_h denote the set of honest and attack-free vehicles. The resilient traffic state estimator in Algorithm 1 (which utilizes the SAT-based scheme) returns an estimate $\hat{v}_{S^*}(t)$ such that:*

$$\left(\frac{1}{N}\sum_{t=t_1}^{t_1+N-1}\mathbb{E}\left(\hat{v}_{S^*}(t) - v(t)\right)\right) \leq \left(\frac{1}{N}\sum_{t=t_1}^{t_1+N-1}\mathbb{E}\left(\hat{v}_{S_h}(t) - v(t)\right)\right).$$

That is, while the legacy sensors are subject to noise η with unknown and possibly large variance (and hence the estimate $\hat{v}^l(t)$ can be far away from the ground truth), Algorithm 1 can select a subset of vehicles whose information is the closest possible, in expectation, to the ground truth velocity.

V. Case Study: Bologna City

To validate our proposed method, we conducted experiments using real traffic data from the Italian city of Bologna, using the Bologna Ringway dataset [24], [37], [38]. This dataset covers a typical day's traffic between 8:00 am and 9:00 am (rush hour), with more than 22000 vehicles; the corresponding heat map, showing typical congestion regions, is given in Figure 3. For our experiments, we follow the Simulation of Urban MObility (SUMO) [39] based simulation methodology for the Bologna dataset [37], [38]. An interesting feature of the dataset is that, during rush hour, commuters mostly drive from residential areas (outside the ring) towards different parts of downtown (inside the ring) where offices and commercial spaces are located [38]. Most of the traffic flow along the ringway since commuters use the fast-transit ringway and enter the inner part of the city when they are close to their destination. In our experiments, we focus on routes whose start and end points are at different

Figure 3. Map of Bologna city traffic system using a heat map showing typical congestion regions during rush hours.

sides of the ringway[1] to allow cars to use a combination of roads inside the ring as well as the ringway.

Within the SUMO framework, we randomly sample cars (at the starting point) and assign them the ‘dishonest’ label (sampling probability varies from one experiment to other as described below). These dishonest cars are then allowed to launch orchestrated Sybil attacks in which a chain of Sybil cars are reporting their presence. The velocity information of the Sybil cars are generated to be consistent with those of dishonest cars. To conduct our experiments, we collected real-time velocity and position data from all cars (dishonest and Sybil cars could report incorrect values) and processed it using our proposed estimation algorithms to produce traffic estimates every one minute. We also introduce one additional car each minute to the original traffic dataset and uses the estimated traffic state to compute optimal route which is relayed back to the SUMO simulator. Finally, these routing decisions are applied to each of the injected cars.

The previous process (collect data, estimate traffic flow, and route cars) was repeated every minute for the duration 8:00 am to 9:00 am. In an ideal scenario with no attacks, the routing algorithm would recommend the route with the shortest travel time in the aforementioned simulation setup. The same experiments are executed using (i) non-secure traffic estimation which averages out the information from all the cars in each sector, (ii) the proposed resilient traffic estimator which uses the SAT-based scheme, (iii) and the proposed resilient traffic estimator which uses the majority-voting scheme. It is the resilience to attacks using our proposed methods that we validate through our experiments described below.

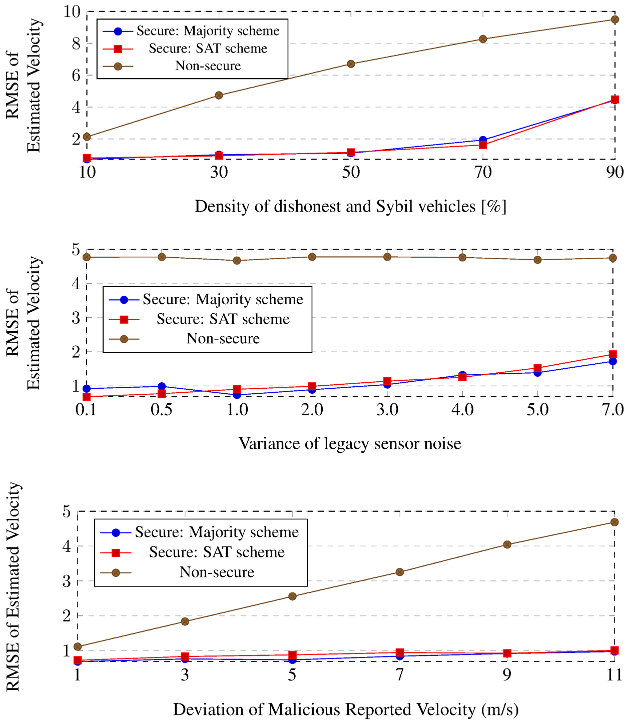

Figure 4. Velocity Estimator Accuracy (top) versus different dishonest/Sybil car probability (middle) versus different legacy sensor error deviation and (bottom) versus attack power.

A. Experiment 1: Accuracy of the Traffic Resilient Estimator

We start by studying the estimation quality, measured by the root mean square error, of the proposed estimator. In particular, we are interested in studying the performance against several attack parameters namely (i) the percentage of dishonest and Sybil cars, (ii) noise in the legacy sensors, and (iii) the attack signal magnitude.

To test the performance of our estimator, we generated scenarios with increasing number of dishonest and Sybil cars, leading to a total density of dishonest and Sybil cars, in the attacked sector, varying between 0% to 70%. Each dishonest or Sybil car is reporting a velocity which is less than the actual velocity by $1m/s$. Figure 4 (top) reports the root mean square error of the velocity estimation, concerning the ground truth collected from the SUMO simulator, averaged over the whole simulation time. The figure shows a monotonic sensitivity of the non-secure estimator concerning the increase in the number of dishonest and Sybil cars. This is a direct consequence of averaging out the information collected from all the cars without differentiating between honest and malicious vehicles. The results also show that both the SAT-based scheme and the majority-based scheme lead to a comparable performance regarding estimation error. This, in turn, supports the heuristic employed by the majority-voting scheme. Also, we note that when the density

[1] Regarding the actual (latitude, longitude): the start point is (44.500915, 11.330234) and the end point is (44.498994, 11.338538). The length of routes is in the 1-1.5 km range.

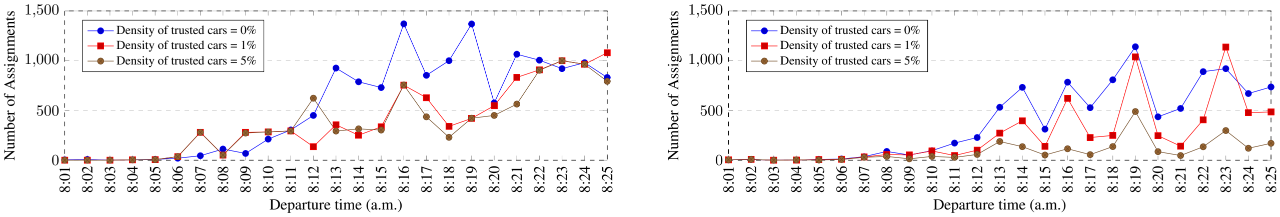

Figure 5. Cardinality of the set $\mathbb{S}_k$ (i.e., number of subsets of consistent cars) for an increasing number of trusted cars (left) using SAT-based scheme and (right) majority-voting based scheme.

of the honest cars decreases, as a result of the increase in the density of dishonest vehicles, the proposed resilient estimator is not able to find enough vehicle measurements at each sector leading to an increase in the root mean square error.

Next, we consider the effect of increasing the noise level in the legacy sensors in Figure 4(middle). Since the non-secure estimator does not take into account the data collected from the infrastructure, the estimation quality of the non-secure estimator is not affected by the noise level in the legacy sensors. On the other hand, when the noise level in the legacy sensors increases, the quality of the proposed estimator slightly decreases. Nevertheless, the proposed estimator is still capable of distinguishing honest cars and hence provides a better estimate of the traffic velocity compared to the non-secure estimator. Again, we notice a comparable performance between both the SAT-based scheme and the majority-based scheme.

Finally, we report in Figure 4 (bottom) the performance results when the malicious (dishonest and Sybil) vehicles increase their attack level by reporting velocities further away from the ground truth velocity. On one side, as the reported velocities deviate more from the ground truth, the quality of the non-secure estimator reduces significantly. On the other side, the proposed resilient estimator is less sensitive to the attack level thanks to its ability to select the subset of cars that report the best traffic information.

B. *Experiment 2: Number of consistent sets*

In this experiment, we report on the number of combinations (subsets of vehicles) that are produced by the SAT solver, i.e., the cardinality of the set $\mathbb{S}_k$. Figure 5 shows the number of subsets of vehicles for both the SAT-based and the majority-voting based schemes (averaged across all sectors) at each minute of the simulation interval. As expected, the number of combinations generated by the majority-voting based scheme is consistently smaller than those generated by the SAT-based scheme.

Next, we consider how the number of subsets of vehicles is affected by the existence of trusted cars. Recall that whenever a trusted car is present, we force the SAT solver to restrict its search to these subsets of vehicles whose information is consistent with those reported by the trusted cars. This leads to examining a smaller number of combinations. This fact is reflected in the results shown in Figure 5. In particular, these results show that even a small number of trusted cars ($1\% - 5\%$) in the whole traffic system can lead to a significant decrease in the number of subsets of vehicles "primarily" in the majority-voting scheme. This decrease in the number of subsets of vehicles, in turn, affects the scalability of the proposed schemes.

C. *Experiment 3: Average Travel Time*

In this experiment, we consider the performance of the whole traffic system measured by the average travel time for the injected vehicles. The recorded average travel time is reported in Figure 6 for two attacker scenarios namely (i) medium-density of dishonest/Sybil cars and (ii) high-density of dishonest/Sybil cars.

In the medium attack scenario, the density of Sybil cars is 30% of the total number of cars reported to the routing algorithm. This attack is not capable of creating a complete "virtual" congestion on the highway. Therefore, the non-secure system routes some of the injected vehicles through the highways (which have light traffic conditions during that time) and the rest of the routed vehicles to the sideways leading to massive traffic on the sideways and a corresponding increase in the average travel time. On the contrary, we note that the proposed resilient traffic estimators can correctly estimate the state of the traffic system and route the added vehicles to the highway which has a light traffic condition (until 8:23 am when the traffic congestions on the highways start to increase). We note that the majority-voting based scheme lead, in general, to a comparable performance compared to the SAT-based scheme except for the time 8:06-8:13 am.

On the other hand, in the high-level density attack scenario (where the density of Sybil cars is 80% of the total number of cars reported to the routing algorithm), the attacker is capable of creating a complete "virtual" congestion on the highway. This high congestion is enough to enforce the non-secure system to route *all* the injected vehicles to the sideways creating actual congestion on the sideways. This, in turn, leads to a considerable high average travel time equal to almost one hour. On the other side, we note that the SAT-based scheme, except for few instances, can correctly

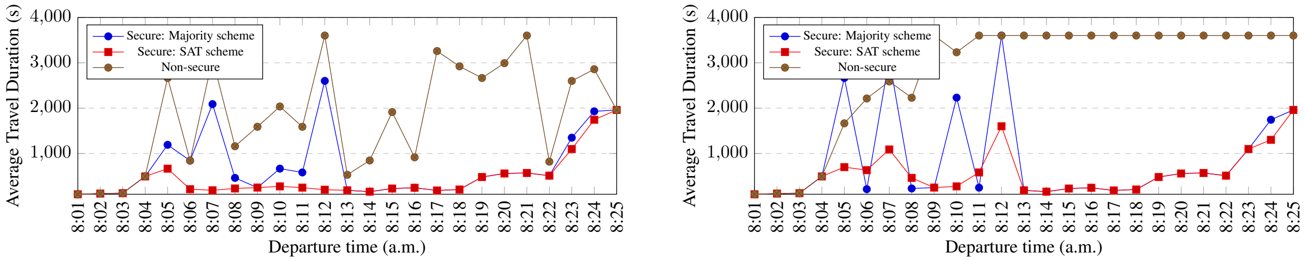

Figure 6. Results showing the average travel time for routed cars using the proposed scheme (left) medium density dishonest/Sybil attack scenarios and (right) high-density dishonest/Sybil attack scenarios.

estimate the highway congestion leading to maintaining the same performance regardless of the density of the attacked vehicles. We also note that the performance of the majority-voting scheme degrades compared to the medium-density attack scenario reflecting the fact that the heuristic, unlike the SAT-based scheme, is not guaranteed always to find the best subset of cars.

VI. Conclusions

In this paper, the problem of estimating the state of the traffic system from maliciously corrupted crowdsourced information is considered. Attackers are assumed to be able to report malicious data and report the presence of Sybil cars which do not physically exist in the traffic network. We proposed a physics-based trust propagation scheme in which the unreliable and sporadically available information from the legacy sensors are used as a root-of-trust to sanitize the crowdsourced information. The result is an estimation algorithm that is resilient to such attacks while being able to compute the optimal estimate of the traffic state. The proposed scheme is analyzed to show optimality and verified using real traffic data collected from the Italian city of Bologna. Simulation results show that our scheme can reduce the average travel time during rush hour from an hour to a few minutes.

References

[1] Waze, “Waze: Free community-based gps, maps and traffic navigation app,” 2017. [Online]. Available: https://www.waze.com/

[2] N. Stern, “Waze’s Drive Towards Successful Public Partnerships,” Data-Smart City Solutions, Tech. Rep., Feb. 2016. [Online]. Available: http://datasmart.ash.harvard.edu/news/article/wazes-drive-towards-successful-public-partnerships-786

[3] Research and Markets, “Semi-Autonomous and Autonomous Vehicles Market by Technology, Components, Powertrain and Region - Global Forecast 2021-2030,” Research and Markets, Tech. Rep., May 2017. [Online]. Available: https://www.researchandmarkets.com/research/3jw44d/semiautonomous

[4] I. Efrati, “Waze under attack: Israeli students fake traffic jam on popular map app,” Mar. 2014. [Online]. Available: https://www.haaretz.com/israel-news/.premium-1.581732

[5] M. B. Sinai, N. Partush, S. Yadid, and E. Yahav, “Exploiting social navigation,” *arXiv preprint*, 2014. [Online]. Available: https://arxiv.org/abs/1410.0151

[6] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao, “Defending against sybil devices in crowdsourced mapping services,” in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Jun. 2016, pp. 179–191.

[7] J. R. Douceur, “The sybil attack,” in *Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, January 2002, pp. 251–260. [Online]. Available: https://www.microsoft.com/en-us/research/publication/the-sybil-attack/

[8] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, “Aiding the detection of fake accounts in large scale social online services,” in *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, San Jose, CA, 2012, pp. 197–210.

[9] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, “An analysis of social network-based sybil defenses,” *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 363–374, 2010.

[10] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, “Sybilguard: defending against sybil attacks via social networks,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 267–278, 2006.

[11] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, “Sybillimit: A near-optimal social network defense against sybil attacks,” in *Proceedings of the IEEE Symposium on Security and Privacy*, 2008, pp. 3–17.

[12] G. Danezis and P. Mittal, “Sybilinfer: Detecting sybil nodes using social networks,” in *Proceedings of the 16th Annual Network & Distributed System Security Symposium (NDSS)*, San Diego, CA, 2009, pp. 1–15.

[13] S. Saroiu and A. Wolman, “Enabling new mobile applications with location proofs,” in *Proceedings of the 10th ACM workshop on Mobile Computing Systems and Applications (HotMobile)*, 2009, p. 3.

[14] W. Luo and U. Hengartner, “Proving your location without giving up your privacy,” in *Proceedings of the 11th ACM Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2010, pp. 7–12.

[15] J. Brassil, P. K. Manadhata, and R. Netravali, “Traffic signature-based mobile device location authentication,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 9, pp. 2156–2169, 2014.

[16] S. Gil, S. Kumar, M. Mazumder, D. Katabi, and D. Rus, “Guaranteeing spoof-resilient multi-robot networks,” *Autonomous Robots*, vol. 41, no. 6, pp. 1383–1400, 2017.

[17] F. Pasqualetti, F. Dorfler, and F. Bullo, “Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems,” *IEEE Control Systems*, vol. 35, no. 1, pp. 110–127, 2015.

[18] H. Fawzi, P. Tabuada, and S. Diggavi, “Secure estimation and control for cyber-physical systems under adversarial attacks,” *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1454–1467, June 2014.

[19] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. Pappas, “Robustness of attack-resilient state estimators,” in *Proceeding of the ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, Apr. 2014, pp. 163–174.

[20] Y. Mo, J. Hespanha, and B. Sinopoli, “Resilient detection in the presence of integrity attacks,” *IEEE Transactions on Signal Processing*, vol. 62, no. 1, pp. 31–43, Jan 2014.

[21] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, “Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach,” *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4917–4932, 2017.

[22] S. Mishra, N. Karamchandani, P. Tabuada, and S. Diggavi, “Secure state estimation and control using multiple (insecure) observers,” in *Proceedings of IEEE Conference on Decision and Control (CDC)*, Dec. 2014, pp. 1620–1625.

[23] S. Mishra, Y. Shoukry, N. Karamchandani, S. N. Diggavi, and P. Tabuada, “Secure state estimation against sensor attacks in the presence of noise,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 49–59, March 2017.

[24] L. Bedogni, M. Gramaglia, A. Vesco, M. Fiore, J. Haerri, and F. Ferrero, “The bologna ringway dataset,” 2015. [Online]. Available: http://www.cs.unibo.it/projects/bolognaringway/

[25] Y. Wang and M. Papageorgiou, “Real-time freeway traffic state estimation based on extended kalman filter: a general approach,” *Transportation Research Part B: Methodological*, vol. 39, no. 2, pp. 141–167, 2005.

[26] J. Reilly, S. Martin, M. Payer, and A. M. Bayen, “Creating complex congestion patterns via multi-objective optimal freeway traffic control with application to cyber-security,” *Transportation Research Part B: Methodological*, vol. 91, pp. 366–382, 2016.

[27] Y. Shoukry, P. D. Martin, P. Tabuada, and M. B. Srivastava, “Non-invasive spoofing attacks for anti-lock braking systems,” in *Workshop on Cryptographic Hardware and Embedded Systems*, ser. G. Bertoni and J.-S. Coron (Eds.): CHES 2013, LNCS 8086. International Association for Cryptologic Research, 2013, pp. 55–72.

[28] Y. Shoukry and P. Tabuada, “Event-triggered state observers for sparse sensor noise/attacks,” *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2079–2091, 2016.

[29] Microsoft Azure website, “Announcing the Coco framework for enterprise blockchain networks.” [Online]: https://azure.microsoft.com/en-us/blog/announcing-microsoft-s-coco-framework-for-enterprise-blockchain-networks/, 2017.

[30] A. Jøsang, S. Marsh, and S. Pope, “Exploring different types of trust propagation,” in *Proceedings of the International Conference on Trust Management (iTrust)*, vol. 3986, 2006, pp. 179–192.

[31] R. Sailer, X. Zhang, T. Jaeger, and L. Van Doorn, “Design and implementation of a tcg-based integrity measurement architecture.” in *Proceedings of USENIX Security Symposium*, vol. 13, 2004, pp. 223–238.

[32] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 135–142.

[33] L. Wu, Y. Zhang, and F. Wang, “A new provably secure authentication and key agreement protocol for SIP using ECC,” *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 286–291, 2009.

[34] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, “Traffic state estimation on highway: A comprehensive survey,” *Annual Reviews in Control*, vol. 43, pp. 128–151, 2017.

[35] Y. Wang, M. Papageorgiou, and A. Messmer, “Real-time freeway traffic state estimation based on extended kalman filter: A case study,” *Transportation Science*, vol. 41, no. 2, pp. 167–181, 2007.

[36] Y. Shoukry, S. Mishra, Z. Luo, and S. Diggavi, “Sybil attack resilient traffic networks: A physics-based trust propagation approach,” *arXiv preprint*, 2018.

[37] L. Bieker, D. Krajzewicz, A. Morra, C. Michelacci, and F. Cartolano, “Traffic simulation for all: a real world traffic scenario from the city of bologna,” in *Modeling Mobility with Open Data*. Springer, 2015, pp. 47–60.

[38] L. Bedogni, M. Gramaglia, A. Vesco, M. Fiore, J. Haerri, and F. Ferrero, “The bologna ringway dataset: improving road network conversion in SUMO and validating urban mobility via navigation services,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5464–5476, 2015.

[39] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO - Simulation of Urban MObility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.