Exposing Congestion Attack on Emerging Connected Vehicle based Traffic Signal Control

Qi Alfred Chen, Yucheng Yin, Yiheng Feng, Z. Morley Mao, Henry X. Liu
University of Michigan
{alfchen, yyucheng, yhfeng, zmao, henryliu}@umich.edu

***Abstract*—Connected vehicle (CV) technology will soon transform today's transportation systems by connecting vehicles and the transportation infrastructure through wireless communication. Having demonstrated the potential to greatly improve transportation mobility efficiency, such dramatically increased connectivity also opens a new door for cyber attacks. In this work, we perform the first detailed security analysis of the next-generation CV-based transportation systems. As a first step, we target the USDOT (U.S. Department of Transportation) sponsored CV-based traffic control system, which has been tested and shown high effectiveness in real road intersections. In the analysis, we target a realistic threat, namely CV data spoofing from one single attack vehicle, with the attack goal of creating traffic congestion.**

We first analyze the system design and identify data spoofing strategies that can potentially influence the traffic control. Based on the strategies, we perform vulnerability analysis by exhaustively trying all the data spoofing options for these strategies to understand the upper bound of the attack effectiveness. For the highly effective cases, we analyze the causes and find that the current signal control algorithm design and implementation choices are highly vulnerable to data spoofing attacks from even a single attack vehicle. These vulnerabilities can be exploited to completely reverse the benefit of the CV-based signal control system by causing the traffic mobility to be 23.4% worse than that without adopting such system. We then construct practical exploits and evaluate them under real-world intersection settings. The evaluation results are consistent with our vulnerability analysis, and we find that the attacks can even cause a blocking effect to jam an entire approach. In the jamming period, 22% of the vehicles need to spend over 7 minutes for an original half-minute trip, which is 14 times higher. We also discuss defense directions leveraging the insights from our analysis.

I. Introduction

Connected vehicle (CV) technology will soon transform today's transportation systems. In September 2016, the USDOT (U.S. Department of Transportation) launched the CV Pilot Program as a national effort to deploy, test, and operationalize a series of CV-based transportation systems [12], [2]. In these systems, vehicles and infrastructure are connected through wireless communication, and leverage such connectivity to improve mobility, safety, environmental impact, and public agency operations. These systems are currently under testing in three cities including New York City [12]. To push for a nationwide deployment, USDOT has already proposed to mandate all new light-duty vehicles to equip CV technology [11].

While having a great potential, such dramatically increased connectivity also opens a new door for cyber attacks. To ensure the security of vehicles and transportation infrastructure and the safety of drivers and pedestrians, it is highly important to understand potential security vulnerabilities so that they can be proactively addressed before nationwide deployment.

In this work, we perform the first security analysis on the next-generation CV-based transportation systems. As a first step, we target the USDOT sponsored design and implementation of a system called Intelligent Traffic Signal System (I-SIG), which performs one of the most basic urban traffic operations, traffic signal control. In this system, real-time vehicle trajectory data transmitted using the CV technology are used to intelligently control the duration and sequence of traffic signals. Such system is fully implemented and has been tested on real road intersections in Anthem, AZ, and Palo Alto, CA, and has shown to achieve a 26.6% reduction in total vehicle delay [6]. In this study, our goal is to identify fundamental security challenges, especially those specific to CV-based traffic control. Thus, we are particularly interested in security problems that are at the signal control algorithm level and are caused by design or implementation choices instead of implementation bugs. The analysis results are expected to serve as a guideline for understanding whether and why the current design or implementation choices in the I-SIG system are vulnerable, as well as providing insights on how to fundamentally secure it before large-scale deployment.

The only attack requirement in our study is that attackers can compromise the vehicle-side devices on their own vehicles or other people's vehicles, and send malicious CV messages to the I-SIG system to influence the traffic control decisions. As reported by previous work, such compromise can be performed physically [29], wirelessly [20], or through malware [34]. Also, we assume that the vehicle certificate system developed by USDOT (§II-A) can correctly authenticate all CV messages. Thus, instead of the sender identity, the attack vehicle can only spoof its trajectory data, e.g., speed and location, in the CV messages. To maximize the realism, in this paper we assume that *only one attack vehicle* exists in an intersection. This ensures that both our analysis and the discovered security problems have high practical implications.

With such a threat model, the attack goal in our analysis is to create congestion in an intersection. Traffic signal control

Network and Distributed Systems Security (NDSS) Symposium 2018
18-21 February 2018, San Diego, CA, USA
ISBN 1-891562-49-5
http://dx.doi.org/10.14722/ndss.2018.23222
www.ndss-symposium.org

has been proven to be one of the most cost effective way to increase transportation productivity, and thus it is highly important to ensure its correct and efficient functioning. This is exactly the reason why the USDOT focuses on deploying the CV-based signal control system [12]. Thus, as the first security study, this work focuses on identifying the congestion creation vulnerabilities, aiming at directly subvert such design goals.

We first analyze the I-SIG system design and identify a set of trajectory data spoofing strategies that can potentially influence the signal control algorithms used in the system. We then enumerate all the data spoofing options for the identified strategies on the I-SIG system to understand the upper bound of the congestion attack effectiveness. A commercial-grade traffic simulation software, PTV VISSIM [8], is used to generate synthetic traffic snapshots as the input to the I-SIG system for this analysis. Based on the results, we analyze the causes for the highly effective attack results, and construct practical exploits under real-world attack resource constraints.

In our analysis, we find that data spoofing attacks are highly effective for the signal control algorithm with the default configurations in I-SIG: the spoofed trajectory data from one single attack vehicle is able to increase the total delay by as high as 68.1%, which completely reverses the benefit of using the I-SIG system (26.6% decrease) and cause the mobility to be even 23.4% worse than that without using the I-SIG system. This is very surprising, since the I-SIG system uses an optimal signal control algorithm that can minimize the total delay of typically over 100 vehicles in an intersection. Thus, the data from a single vehicle should not have such significant influence. We find that this is due to a vulnerability at the signal control algorithm level, which we call *the last vehicle advantage*, meaning that the latest arriving vehicle can determine the signal plan. Fundamentally, we find that this is due to a trade off between security and deployability: due to the limited computation power on the infrastructure-side devices, the developers are forced to choose a less optimal configuration of the theoretically optimal signal control algorithm, which unexpectedly exposes the congestion creation vulnerability.

Even though the deployability issue exists today, this problem may be resolvable in the future when the infrastructure-side devices have more computation power. Thus, we then analyze whether the I-SIG system is still vulnerable with more optimal configurations. In such scenario, we find that data spoofing attacks can still be highly effective when the I-SIG system is in the operation mode for the transition period, i.e., when the market penetration rate (PR) of the CV technology is lower than 95%. In such period, an algorithm that estimates the status of unequipped vehicles, i.e., vehicles without CV technology, is performed before the signal control algorithm. This is because the signal control algorithm can be very ineffective due to lack of visibility of the unequipped vehicles, but we find that this allows the attacker to manipulate such estimation process to create congestion using spoofed data.

To understand the real-world implications of the identified vulnerabilities, we construct and fully implement the exploits, and evaluate them using simulations on a real-world intersection map. To increase the realism, we videotaped all traffic flows in the intersection for one hour and manually counted the passing vehicles as the input to the simulation model. The results are consistent with our vulnerability analysis, and surprisingly, we find that the attacks can even cause a blocking effect to jam an entire approach. Fig. 1 shows an snapshot in the simulation when the blocking effect is created. As shown, in the northbound, southbound, and eastbound approaches, the vehicles in the left-turn lanes spill over and block the through lanes, causing massive traffic jams. In such jamming period, 22% of the vehicles need to spend over 7 minutes for an originally half-minute trip, which is 14 times higher.

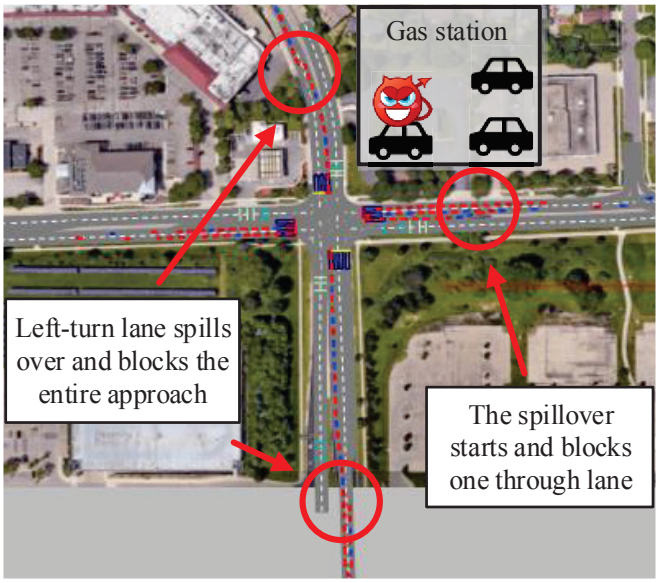

Fig. 1: The blocking effect created by our congestion attack on a real-world intersection map with real traffic demand. Due to the attack from *one single attack vehicle* parking nearby, in the northbound and southbound approaches the vehicles in the left-turn lanes spill over their lanes and directly block the entire approaches, causing massive traffic jams.

Based on our analysis, even though the I-SIG system has shown high effectiveness in reducing traffic delay in benign settings, the current algorithm design and configuration choices are high vulnerable to data spoofing, and even the data from one single attack vehicle is able to manipulate the traffic control to a great extent, causing massive congestion. To addressed these problems, we discuss promising defense directions leveraging the insights from our analysis.

We summarize our contributions as follows:

- We perform the first security analysis of a CV-based transportation system, the USDOT sponsored I-SIG system. We formulate the problem with a highly realistic threat model, data spoofing from one single attack vehicle, and analyze the system design to identify a set of data spoofing strategies.

- Targeting the goal of creating congestion, we first perform vulnerability analysis to understand the upper bound of the attack effectiveness. We analyze the causes for the highly effective cases, and find that the current signal control algorithm design and configuration choices are highly vulnerable to data spoofing from even a single attack vehicle. These vulnerabilities exist throughout the full deployment and the transition periods, and can cause the mobility to be even worse than that without using the I-SIG system.

- For the identified vulnerabilities, we construct practical exploits and evaluate them under real-world intersection settings. The results validate the attack effectiveness; furthermore, for the transition period, the attacks can even create a blocking effect that jams an entire approach.

II. Background

In this section, we introduce the necessary background about the CV technology and the I-SIG system design.

A. CV Technology and Recent Advances

Connected vehicle (CV) technology uses wireless communications to connect vehicles and the infrastructure with the goal of dramatically improving the transportation systems in mobility, safety, environmental impact, and public agency operations [12]. Due to the high data transmission requirement in the transportation scenario, the DSRC (Dedicated Short Range Communications) protocol is specifically designed for the CV communication scenarios with dedicated band allocated by the Federal Communications Commission (FCC) [5].

The communication in the CV environment has two categories: vehicle-to-vehicle (V2V) communication, and vehicle-to-infrastructure (V2I) communication. To support them, both the vehicle and the infrastructure sides need to install DSRC devices, which are called On-Board Units (OBUs) and Roadside Units (RSUs) respectively. In such CV environment, vehicles use OBUs to periodically broadcast Basic Safety Messages (BSM) including its real-time trajectory data, e.g., location and speed, to the surrounding vehicles and infrastructure. This enables a series of safety functions on the vehicle side, e.g., blind spot and lane change warnings, and also enables the traffic infrastructure to leverage the real-time traffic data to improve traffic control performance.

Recent advances in the CV deployment. With the DSRC standard becoming mature [28], OBUs and RSUs products are already on market today [1]. USDOT estimates that equipping the OBUs would cost around $341 to $350 per vehicle in 2020 [13]. This makes the CV technology a very cost-effective option to increase transportation system performance in practice, and the USDOT has already proposed to mandate all new light-duty vehicles to equip OBUs [11]. The market penetration rate will gradually increase after such mandate [16], and in our analysis we call the vehicles with and without OBUs *equipped vehicles* and *unequipped vehicles* respectively.

To foster the development of CV-based transportation systems, in 2010 the USDOT launched the Dynamic Mobility Applications (DMA) research program and developed nearly 70 such systems, or CV applications [2]. To encourage service providers, researchers, and application developers to participate, these applications are open sourced and are available free to the public [7]. Built on the success of the DMA program, on September 1, 2016, the USDOT awarded $45 million to start small-scale deployment of these systems, called the CV Pilot Deployment Program, in three sites including New York [12]. In this paper, we perform the first security analysis of such CV-based transportation systems as a timely study to understand the potential security problems and challenges at the design level before large-scale deployment.

Security and Credential Management System (SCMS). As one of the most importatnt infrastructure, the transpiration systems are highly security and safety critical. Thus, to enhance the communication security in the CV environment, the USDOT will deploy the Security and Credential Management System (SCMS) on both the vehicle and infrastructure sides [15]. It is a Public-Key Infrastructure (PKI) system that requires every BSM messages to be signed by the sender's digital certificates issued beforehand, and thus the receivers can verify the signature before acting on it [15], [38].

B. The I-SIG System

As the first security study on CV-based transportation systems, we target the CV-based traffic control system developed in the DMA program, called Intelligent Traffic Signal System (I-SIG) [14]. In this system, real-time vehicle trajectory data transmitted via DSRC are leveraged to perform more effective traffic signal control in an intersection.

In the DMA program, the development of I-SIG was assigned by USDOT to a team of signal control experts. In this work, we use the latest released version, MMITSS-AZ [4]. This version is fully functional in the field, which has been tested in real intersections in Anthem and Palo Alto and shown high effectiveness with a 26.6% reduction in the total vehicle delay [6]. In this section, we first introduce some key concepts in signal control, and then describe the I-SIG system design.

1) Traffic Control Concepts: As shown in Fig. 2, the I-SIG system is operated on an RSU located at an intersection to control the traffic signals. As shown, there are 8 traffic signals, called *phases*. Phases with odd numbers are for left-turn lanes; the others are for through lanes. Each phase is initially configured with the minimum green light lasting time, $t_{g_{min}}$, the maximum green light lasting time, $t_{g_{max}}$, the yellow light lasting time t_y, and the clearance red light lasting time t_r. During the signal control, a phase can be set to turn green and last for a duration t_g. The green duration t_g must be at least $t_{g_{min}}$ and at most $t_{g_{max}}$; this is enforced at the hardware level. After t_g time passes, the phase will be yellow for t_y, and then red for t_r before the subsequent phase turns green, which is for safety purposes since there might be red light runners.

Signal control is performed by setting t_g and the phase sequence, which in combination called a *signal plan*. Fig. 3 illustrates a signal plan. Number 1 to 8 are phases, and the green, yellow, and clearance red light periods for each phase are filled with the corresponding colors. As shown, this plan has two phase sequences, called *rings*, operating simultaneously. The phases in the same ring is in conflict with each other, and thus need to be planned sequentially. This is called dual-ring signal phasing, which is the NEMA (National Electrical Manufacturers Association) standard and the most common in the U.S. [37]. For each ring, the phase sequence is broken down to stages. Two types of stages are planned alternatively, one for phase 1, 2, 5, and 6, and another for phase 3, 4, 7, and 8. The phases in the former stage type are in conflict with those in the latter stage type, and thus the phases in the same stage are planned as a whole.

A signal control algorithm needs to follow the rules above, and plan (1) t_g for each phase, and (2) the sequence of phases in the same ring and same stage, e.g., phase 1 and 2 in the

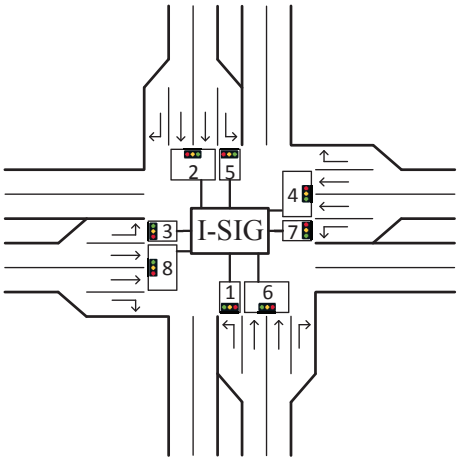

Fig. 2: The operation scenario for the I-SIG system.

Fig. 3: Illustration of a signal plan. Number 1 to 8 are phases.

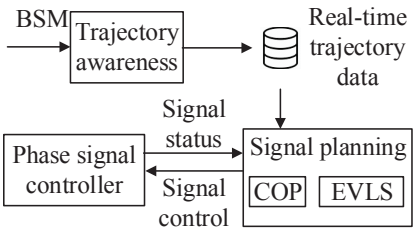

Fig. 4: The I-SIG system design.

figure. A typical goal of such algorithm is to reduce the total delay for all vehicles in the intersection. The delay time for a vehicle spent in an intersection is calculated as the actual time the vehicle spent to pass the intersection subtracting the so-called *free-flow* travel time, meaning that the vehicle is traveling at the speed limit without slowing down or stopping due to red lights or other vehicles. The traffic load for an intersection is called *traffic demand*.

2) System Design: Fig. 4 shows the design of the I-SIG system. The BSM messages broadcast by the equipped vehicles are received by a component called trajectory awareness, which maintains the latest trajectory for each vehicle indexed by the vehicle ID in the BSM messages. It also does some pre-processing tasks for the use in the signal planning component, e.g., assigning vehicle data to their requested phases based on the intersection map. The signal planning component listens to the traffic signal status reported by the signal controller, and launches signal planning stage by stage. More specifically, at the beginning of each stage, the signal planning component pulls the pre-processed real-time trajectory data for the vehicles in the intersection, performs the planning, and sends signal control commands to the signal controller. In the current design, the following algorithms are used for signal planning:

The COP algorithm. The signal planning in the I-SIG system uses a dual-ring version of the COP (Controlled Optimization of Phases) algorithm[36], [25]. The input of the COP algorithm is each approaching vehicle's estimated arrival time at the intersection, which is defined as the estimated remaining time for a vehicle to reach the stop bar of its current lane. Based on the arrival time, COP uses dynamic programming to calculate an optimal signal plan with the least estimated total delay. To estimate the total delay, COP first estimates the releasing time for a vehicle based on the queue length at its arrival time. If there is no queue, there is no delay; otherwise, it uses a queuing model to estimate when the queue before the vehicle is cleared. Then, the delay for a vehicle is calculated as its releasing time subtracting its arrival time. If there are no vehicle requesting a certain phase, COP skips this phase in its planning so that the subsequent phases that have vehicle request can be planned earlier.

In the design, COP can plan for unlimited number of stages until all vehicles in the intersection can be served based on its estimation. Since there might be more vehicles arriving at the phases in the second stage, the I-SIG system only applies the planned signal duration for first stage at each signal control time. Since the operation of the signal controller requires to know what the next phase is after the current phase, the I-SIG system also sets the phase sequence for the next stage at the time of signal control. This means that in Fig. 3, the I-SIG system cannot change the order of phases in the first stage, since this is set by the signal control last time. It can change the duration of these phases, and the sequence of the phases in the second stage based on the output of the COP algorithm.

In the current I-SIG system, a limit of the planning stages is configured in COP. This is because in practice the signal planning needs to finish within $t_{g_{min}}$, usually 5-7 seconds, in order to be applied to the signal controller in time. Thus, with computation and memory resource constraints in practice, COP cannot plan with unlimited stages like in its design. With limited planning stages, the COP algorithm may not be able to serve all vehicles. Thus, the current implementation in the I-SIG system first finds the plans with the least unserved vehicles, and then choose the one with the least total delay. As shown later in VI-B, such planning stage limit unexpectedly leaves the I-SIG system vulnerable to congestion attacks.

Transition period: the EVLS algorithm. If the COP algorithm only optimizes the signal plan for the equipped vehicles, its effectiveness is found to be largely reduced if the portion of the equipped vehicles is not sufficiently high, e.g., less than 95% [25]. Since it is estimated that the market penetration rate needs 25-30 years to reach at least 95% [16], the I-SIG system uses an algorithm called EVLS (Estimation of Location and Speed) to estimate the trajectory data of the unequipped vehicles. In the EVLS algorithm, the trajectory data of the equipped vehicles is used for such estimation leveraging multiple traffic models (detailed later in §V-B).

Design representativeness and current deployment. The use of COP and EVLS is chosen by the I-SIG designer, the team of USDOT-selected signal control experts, based on a 2015 paper published in Transportation Research Part C [25], a top-tier journal in transportation research. The COP algorithm is chosen because it is very suitable for the CV environment: its input is the arrival time for individual vehicles instead of aggregated traffic information, and thus can best

leverage the per-vehicle trajectory data in the CV environment to effectively handle traffic dynamics. As discussed earlier, the EVLS algorithm is developed to overcome the limitation of COP in the transition period. To the best of our knowledge, this is the only design in the transportation literature that is fully implemented and tested on real roads. In the CV Pilot Program, this system is currently under deployment in Tampa [3].

III. Threat Model

As illustrated in §II-B, the operation of the I-SIG system involves both infrastructure-side devices, i.e., RSUs and signal controllers, and vehicle-side devices, the OBUs. Previous work found that the traditional transportation infrastructure side devices tend to use weak credentials so that attackers can easily take full control [27]. This is a known problem across many embedded network devices [22] and we assume that the next generation CV-based transportation systems will be fully aware of this problem, and adopt sufficiently strong authentication mechanisms as advised by previous work [27] so that they cannot be easily compromised.

Thus, in this work we focus on the attacks from the vehicle-side devices, the OBUs. More specifically, we assume that the attacker can compromise the in-vehicle systems or OBUs on their own vehicles or others' vehicles so that she can send malicious BSM messages to the RSUs to influence the signal plan. It's important to note that we do not assume that the attackers can spoof the sender identities in the BSM messages. Introduced in §II-A, the USDOT will deploy the SCMS system to ensure that all BSM messages are authenticated. Since in this paper we are more interested in new security problems specific to CV-based traffic control, we assume that the SCMS system is sufficiently tested and not easily exploitable.

Thus, in our threat model the attack vehicles need to use their true identities so that the sent BSM messages are still correctly signed, but send spoofed vehicle trajectory data, e.g., speed and location, in these messages. This can be achieved in two ways. First, the attacker may directly compromise OBUs by exploiting software vulnerabilities, similar to the demonstrated compromises on other Electronic Control Units (ECUs) [29], [20]. Second, if compromising OBUs is difficult, the attacker can send fabricated CAN messages with spoofed sensor data to the OBUs by compromising other ECUs [29], [20], [21]. Since the attack model includes malicious vehicle owners who have *arbitrary* physical accesses, as long as in-vehicle systems are not vulnerability-free, which has been proved repeatedly [29], [20], [34], such compromises are always achievable in practice, just like the smartphone jailbreaking/rooting practices today.

To maximize the realism of our threat model, in this paper we assume that *only one attack vehicle* presents in an intersection. Since the COP algorithm targets optimized total delay for all vehicles in an intersection, which normally have over 100 of them, it should be very challenging for the data from one single vehicle to significantly influence the signal planning. However, as shown later, this is actually possible due to several newly-discovered vulnerable design and configuration choices.

The attacker is assumed to have limited computation power to launch the attack, e.g., only using a consumer laptop. More specifically, when using paralleled computation, the attack laptop is assumed to have four processors to execute simultaneously, which is a common specification for consumer laptops such as Macbook Pro. Before attacking an intersection, the attacker is assumed to have performed sufficient reconnaissance and thus already knows (1) the signal control algorithm choices, by testing the algorithm-specific vulnerabilities identified in this paper (detailed later), and (2) signal control configurations and the intersection map, by measuring the opened phases, the corresponding signal duration, and the intersection map beforehand.

Since in the CV environment the vehicles are broadcasting BSM messages to the surrounding devices (§II-A) and the attack vehicle is in the victim intersection, we assume that the attack vehicle can receive the same set of BSM messages as those in the RSU. Thus, they can run the COP and EVLS algorithms themselves to know the executed signal plans and also estimate the signal plans to be executed, which is also implemented in our exploitation process(§VII).

IV. Analysis Methodology Overview

In this section, we describe the target attack goal and overview the analysis methodology.

A. Attack Goal: Creating Congestion

As the first security study on CV-based signal control, our analysis in this paper focuses on subverting the core design goal of the I-SIG system, total vehicle delay reduction. More specifically, the attacker aims to send spoofed trajectory data to influence the signal plan in order to increase the total delay of other vehicles in the intersection. The attack vehicle is not necessarily in the traffic flows; it might just park nearby, e.g., in a gas station as shown in Fig. 1, listening to the BSM messages from other vehicles, and seek chances to launch attacks.

Attack incentives. Such attacks can be politically or financially incentivized, e.g., blocking routes to business competitors, like denial-of-service attacks on Internet. Since one attack vehicle can only attack one intersection, to cause larger-scale damage, attackers can form groups to attack consecutive intersections along arterial roads in an area.

Damage to city functions and individuals. As one of the critical infrastructure, signal control systems has a fundamental impact on economic and environment, and thus it is highly important to ensure that such system is well protected and functions correctly and efficiently. This is equally true from individual's perspective: as estimated by a recent study, traffic jams cost U.S. drivers an average of $1,200 a year in wasted fuel and time [10]. This is exactly the reason why the USDOT is pushing the deployment of CV-based signal control [12].

B. Analysis Methodology Overview

To understand how vulnerable the current I-SIG system design and implementation is under our threat model, our security analysis consists of the following key steps:

(1) Data spoofing strategy identification. Before analyzing the vulnerability of the I-SIG system, we first need to identify the meaningful data spoofing strategies. Since the attack input is the data in the BSM messages, we analyze the

data flow of the I-SIG system starting from the receiving BSM messages to understand how the spoofed data can potentially influence the signal control.

(2) Vulnerability analysis for each attack goal. With data spoofing strategies identified, we then enumerate all the data spoofing options for these strategies on the I-SIG system to understand the upper bound of the congestion attack effectiveness through data spoofing. To analyze the I-SIG system, we need realistic vehicle trajectory data as input to trigger the signal plan. Since it is impossible to use real vehicles in an intersection due to ethical concerns, our analysis uses a commercial-grade traffic simulation software, PTV VISSIM [8], to simulate traffic patterns with a realistic modelling of driver behaviors.

To ensure the generality of this analysis, we create an intersection map with the a generic intersection structure and the common phase configuration in the U.S. We then use VISSIM to generate traffic flows of normal demand following the common practices in the transportation research area. We take snapshots of the vehicle trajectory data in the simulation periodically, which is then used as the input to our analysis. For each snapshot, we run the signal planning in the I-SIG system with and without attack data input, and quantify the attack effectiveness in creating congestion.

(3) Cause analysis and practical exploit construction. With the attack effectiveness for all possible data spoofing options quantified, we perform cause analysis for the highly effective attacks to understand why the current signal control is vulnerable. Leveraging the insights, we construct practical exploits under real-world attack resource constraints, e.g., computation power of a normal laptop as described in our threat model (§III). As detailed later in §VII, this means that the attacker cannot exhaustively try all possible data spoofing options before making the attack decision; instead, she needs to strategically plan the attack decision process to ensure that she does not miss the attack timing.

(4) Evaluation using simulations with real-world intersection settings. To more concretely understand the practical impact of the constructed exploits, we implement and evaluate these exploits using simulations with real-world intersection settings. We use the map of a real-world intersection with its real phase configurations, and generate traffic flows according to the real traffic demand that we manually measured for one hour on that intersection. Also, compared to attacking individual snapshots in the vulnerability analysis step, in this experiment the attacks are continuously launched for one hour, closely evaluating real-world attack situations.

V. Data Spoofing Strategy

As the first step in our analysis, in this section we analyze attack input data flows to identify data spoofing strategies.

A. Attack Input Data Flow and Direct Spoofing Strategy

Fig. 5 shows the attack input data flow in the I-SIG system. When the spoofed vehicle trajectory data is received, it first performs a geofence check, and only accepts the data if its location is within the geographic boundaries of the intersection. Thus, as described in §III, the attacker needs to perform reconnaissance to know the geographic coordinates of a targeted intersection, and only generate valid location data to pass the geofence check.

Then, if the configured PR in the I-SIG system is lower than 95%, it is considered a transition period and the attack data are feed into the EVLS algorithm to estimate the trajectory data for the unequipped vehicles. Otherwise, it is considered a full deployment period and the EVLS algorithm is skipped.

A list of vehicle trajectory data entries, including the ones for both the equipped vehicles and the estimated unequipped vehicles if it is during the transition period, is then processed to a structure called *arrival table*. An arrival table is an array with two dimensions: the estimated arrival time and the phases. The arrival time is rounded to seconds. Each array element at (i, j) is the number of vehicles for the arrival time i at phase j. The first row is for vehicles with zero arrival time, meaning that they are stopped (speed is 0) and waiting in queue.

The COP algorithm computes a signal plan with the optimal total delay for all vehicles based on the arrival table. Thus, the direct goal of the data spoofing attack is to change the values in the arrival table so that it can influence the planning in the COP algorithm. Since each vehicle has a position in the arrival table, the direct data spoofing strategy is:

S1. Arrival time and phase spoofing, for both the full deployment and transition periods. In both the full deployment and transition periods, the attacker can change the speed and location in its BSM message to set the arrival time and the requested phase of her choice and thus increase the corresponding arrival table element by one. In current implementation, the arrival table considers vehicles arriving in no more than 130 seconds. Thus, in this strategy the attacker has 131 (arrival time) $\times$ 8 (phase) data spoofing options.

B. Spoofing Strategy For The Transition Period Only

To change the arrival table, besides directly spoofing the attack vehicle's own data, the unequipped vehicle estimation process in the transition period is another attractive attack target. Since both the data from equipped and unequipped vehicles are considered in the arrival table, manipulating the estimation results may more significantly influence the signal plan than only changing one vehicle's data in S1.

The unequipped vehicle estimation process, i.e., the EVLS algorithm [25], is detailed in the lower part of Fig. 5. As shown, the equipped vehicle data for each lane are first assigned into three regions: (1) queuing region, including vehicles waiting in the queue with zero speed, (2) slow-down region, including vehicles slowing down because of the front vehicles, and (3) free-flow region, including vehicles far away from the queue so that they behave independently. The algorithm first finds the stopped equipped vehicle that is the farthest from the lane stop bar and uses its location as the end of the queuing region. The slow-down region started right after the queuing region, and the algorithm uses the equipped vehicle's trajectory data to judge whether it is slowing down due to an unequipped front vehicle based on a car-following model. After the slow-down region begins the free-flow region.

After the region assignment, the algorithm first estimates the number of vehicles in queue by dividing the length of the queuing region by the sum of the vehicle length and the

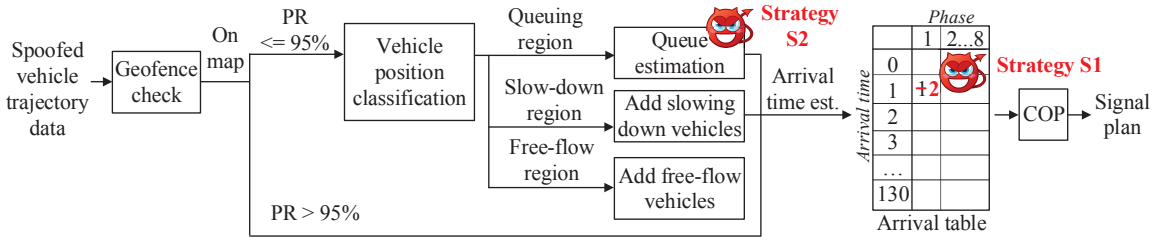

Fig. 5: The data flow of spoofed vehicle driving data in the I-SIG system. PR means penetration rate.

headway in queue, which is 6.56 meters in the implementation. For the slow-down region, for each pair of adjacent equipped vehicles, the algorithm inserts unequipped vehicles between them based on the car-following model. Then if the number of vehicles after the vehicle addition in the queuing and slow-down regions is smaller than the number of equipped vehicles divided by the PR, the algorithm adds the remaining unequipped vehicles to the free-flow region.

Among the three regions, we find that manipulating the estimation of the queuing region is most effective. The attacker can just set the speed to zero and set its location to the farthest possible point of the most empty lane within the geofence so that the lane can be fully filled with queuing vehicles after the estimation. In comparison, attacking the slow-down region is less effective since (1) the number of vehicles it can add is fewer since the space headway between moving vehicles in the car-following model is larger than that between queuing vehicles, and (2) the increased delay by adding moving vehicles is no greater than that by adding queuing vehicles, since the queuing releasing process can create more delay as introduced in §II-B. Since the COP algorithm is designed to optimize the total delay, more vehicles to add and more delay time to increase can have more impact on the signal planning.

Thus, the best strategy is attacking the queue estimation:

S2. Queue length manipulation, for the transition period only. In the transition period, the attacker can change the speed and location data in its BSM message to set the location of the farthest stopped vehicle in a chosen lane, and thus add a number of unequipped queuing vehicles after the original farthest stopped vehicle in the EVLS algorithm. Since this attack only adds queuing vehicles, the change to the arrival table is at the first row. For each phase, the attacker has multiple data spoofing options that can increase the value from by one to by the maximum queue length she can add considering the location of the originally farthest stopped vehicle and the geofence range of the lanes in that phase.

VI. Congestion Attack Analysis

In this section, we use the identified data spoofing strategies to analyze the vulnerability status of the I-SIG system.

A. Experiment Setup

Traffic snapshot generation. As described earlier in §IV-B, we use a generic intersection settings for this analysis. The intersection structure, e.g., number of lanes for each phase, is shown earlier in Fig. 2. The speed limits for all approaches are 40 mph. Each arm of the intersection is set to about 300 meters from the center of the intersection, which is similar to the DSRC communication range [24]. The $t_{g_{min}}$, $t_{g_{max}}$, t_y, and t_r of each phase are configured according to the recommendations from the Signal Timing Manual [37]. In this generic intersection, we use VISSIM to generate vehicles at 0.7 v/c (vehicle per capacity), which corresponds to the medium traffic demand level [33]. Then we run the I-SIG system, and take vehicle trajectory snapshots every time the I-SIG system needs to perform signal planning.

We run the traffic simulation for each scenario three times, each time lasting one hour with a different random seed following the common practices in the transportation research area [36], [19]. In total, we generated 873 snapshots. These snapshots are directly used when we experiment for the fully deployment period. When experimenting for the transition period, we consider three PR levels, 25%, 50%, and 75%, which is the same as that in the EVLS algorithm paper [25]. In these experiments, we still use the 873 snapshots, but randomly select a subset of data according to the PR. The random seed for such selection is the same for all experiments with the same PR so that their results are comparable.

Attack data generation. Using these snapshots, we perform vulnerability analysis of the I-SIG system for congestion attacks by trying all data spoofing options for the strategies identified in §V. For the full deployment period, only strategy S1 is experimented, and for the transition period, both S1 and S2 are experimented. For each data spoofing trial, a new vehicle trajectory data entry with the spoofed data is added to the traffic snapshot as the attack input.

Attack effectiveness analysis. For each snapshot, we run the I-SIG system to get the signal plans with and without attack. Since our goal is to understand the upper bound attack effectiveness, for a snapshot and a chosen data spoofing strategy, we pick the attack result from the most effective data spoofing trial. We analyze the attack effectiveness by comparing the total delay of all vehicles in the snapshot. For the signal plans with attack, the total vehicle delay time is calculated after the attack vehicle data being removed. For the transition period, the ground truth unequipped vehicle data (instead of the estimated data) are used in the calculation.

In the delay calculation, we use the same vehicle delay estimation method in the COP algorithm (§II-B). Since this calculation is based on the arrival time estimation, the calculated delay is not the actual delay since the vehicles may not behave as predicted after the snapshot is taken. However, considering that the COP algorithm has a demonstrated effectiveness [6], [25], such estimation is effective for our purpose, i.e., comparing the attack effectiveness among different attack

trials. In addition, since our goal is to study the vulnerabilities at the signal control algorithm level, using this estimation method allows us to directly evaluate the attack's influence on the signal planning in the COP algorithm. Later in our attack evaluation (§VIII), we will directly measure the actual vehicle delay using the ground truth vehicle trajectory in VISSIM.

In the analysis, we quantify the attack effectiveness using three metrics: (1) attack success rate, which is the percentage of the snapshots with the total delay increased under the attack, which we also call *vulnerable snapshots*, (2) average delay increase time, which is the average absolute increase of the total delay under attack, and (3) average delay increase percentage, which is the average ratio of the increased total delay under attack to the total delay without attack.

B. The Full Deployment Period

In this section, we analyze the attack results for the full deployment period, which are shown in Column 2 to 3 in Table I. In these columns, non-successful attacks means that the total vehicle delay is not changed. As introduced in §II-B, the COP algorithm implemented in I-SIG configures a limit on the number of planning stages. By default it uses two-stage planning, which is denoted as 2-S in the table. We first analyze the results with such default configuration:

1) Two-stage Planning Results: As shown in Column 2 in Table I, we find that S1 is quite effective in creating congestions: it is able to successfully increase the total delay for nearly all (99.9%) snapshots with as high as 68.1% delay increase. In comparison, the benefit of using the I-SIG system is only a 26.6% total delay reduction [6], but our attack can completely reverse such benefit and cause the traffic mobility to be even 23.4% worse than that without using the I-SIG system. This is very surprising, since COP optimizes for the total delay of typically over 100 vehicles in an intersection, and a single vehicle data should not have such significant influence.

Vulnerability cause: last vehicle advantage. By manually examining the signal plan output, we find that for all the vulnerable snapshots, the most successful attack trial adds a spoofed vehicle with very late arrival time. In this paper, we call it the *last vehicle advantage*, which is illustrated in Fig. 6. As shown, in the signal plan, such late vehicle determines the green light end time for its requested phase. This delays the green light begin time for all the phases after it, and thus increases the delay for the vehicles in these phases. If t_g of the phase with this late vehicle reaches $t_{g_{max}}$, the t_g for the phases before this phase will also extend in order to serve this late vehicle, which further delays the vehicles in later phases. Fig. 6 illustrates such attack on phase 2. As shown, due to the spoofed late arriving vehicle, the t_g of all the phases in the first stage are extended in order to be able to serve it, causing long delay to serving time of the vehicles in the second stage.

However, as an algorithm optimizing for the delay of all vehicles, COP should just give up serving this very late vehicle in this green light if serving it costs too much delay for other vehicles. We find that the root cause lies in the planning stage limit when implementing COP in practice. Since the default configuration uses two-stage planning, each phase can only be planned once. Thus, for each phase, planning has to serve all vehicles in this only serving opportunity, causing the planning to be significantly affected by the last arriving vehicle.

This issue can be alleviated when the COP algorithm is allowed to plan for more stages. For example, if the planning stage limit is four, COP now has two opportunities to serve the vehicles for one phase. Thus, even if a vehicle arrives very late, it can delay serving it to the second opportunity. In this case, vehicles in other phases can be served in the first opportunity and thus is less likely to be affected. Fig. 7 shows the percentage of snapshots vulnerable to the last vehicle advantage for the COP algorithm configured with two-stage to eight-stage planning. In the calculation, a snapshot is concluded vulnerable if the most successful attack trial comes from a spoofed vehicle arriving the last in its request phase. As shown, for two-stage planning, nearly all the snapshots can be the most successfully attacked using the last vehicle advantage, and such percentage decreases when more planning stages are configured. The most significant decrease is at four-stage planning, since with such configuration all phases get two serving opportunities. With over four planning stages, the last vehicle advantage is no longer the best trial for any snapshot.

Trade off between security and deployability. Knowing that two-stage planning is highly vulnerable to late arriving vehicles, we are curious why the I-SIG system developers chose to set it as the default value. We contacted the developers and find that it is actually an interesting trade off between deployability and security. As indicated by the developers, they chose two-stage planning because the running time for more planning stages are too high in practice to meet the planning deadline. Since the planning has to finish in $t_{g_{min}}$ (§II), which is typically around 5-7 seconds [37], they told us that running three-stage planning on their RSUs takes more than three seconds due to the limited computation power on RSUs, making it too risky to use. Meanwhile, in their testing, they find two-stage planning does not have much planning effectiveness degradation in comparison to five-stage planning, so they choose it as the default value.

They told us that they use the mainstream Savari StreetWAVE RSU [9] and the 95 percentile running time for two-stage planning takes 1.2 seconds. We then use the ratio between this number and the corresponding running time on our machine to estimate the running time for more planning stages on these RSUs. As shown in Fig. 7, our estimation results are consistent with their observations: purely running COP with three planning stages takes around 3 seconds, and with communication delay and the running time of other parts, e.g., the EVLS algorithm, it is indeed risky to use more than two planning stages. In our snapshots without attack, we also confirmed that using two-stage planning only has 6.5% increase in total delay on average than that using five-stage planning. Thus, choosing two-stage planning is indeed a practical choice that trades small planning effectiveness degradation for reliability. However, such choice is found to be highly exploitable leveraging the last vehicle advantage.

Expected to be mounted outdoor in every intersection, RSUs need to be sufficiently reliable with low cost, which leads to performance constraints just like many real-time embedded systems today [35], [32]. While we have shown that such constraints today cause security vulnerabilities, we envision that this situation may be resolvable in future when

CV deployment	Full deployment		Transition period											
	100% PR		75% PR				50% PR				25% PR			
COP config.	2-S	5-S	2-S		5-S		2-S		5-S		2-S		5-S	
Strategy	S1	S1	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
Vulnerability analysis (exhaustively try all data spoofing options)														
Success %	99.9%	96.4%	99.1%	98.3%	83.2%	96.8%	99.4%	99.2%	83.0%	97.4%	99.9%	98.9%	82.0%	91.6%
Ave. delay	1078.7	162.7	982.2	536.3	167.3	533.9	1001.3	536.2	206.6	569.6	1009.2	531.1	295.8	616.7
inc. (s) & %	68.1%	11.5%	60.2%	32.7%	10.6%	33.5%	61.4%	33.0%	12.5%	34.6%	60.6%	32.4%	17.0%	34.3%
Practical exploit (strategically try data spoofing options due to attack decision time limits in practice)														
Ave. trial #	3.8	13.3	3.8	N/A	N/A	14.7	3.8	N/A	N/A	23.9	3.6	N/A	N/A	28.8
Success %	99.8%	84.7%	99.1%	N/A	N/A	95.6%	99.4%	N/A	N/A	96.6%	99.8%	N/A	N/A	91.5%
Ave. delay	1077.4	119.8	1057.1	N/A	N/A	595.3	1061.0	N/A	N/A	591.7	1008.98	N/A	N/A	609.6
inc. (s) & %	68.0%	9.3%	60.0%	N/A	N/A	35.4%	61.2%	N/A	N/A	35.1%	60.6%	N/A	N/A	33.9%

TABLE I: Vulnerability analysis results and practical exploit effectiveness for congestion attacks. PR is short for penetration rate. Two-stage planning and five-stage planning in the COP algorithm configuration are denoted as 2-S and 5-S respectively, with the former being the default choice. N/A means that practical exploit construction is not performed.

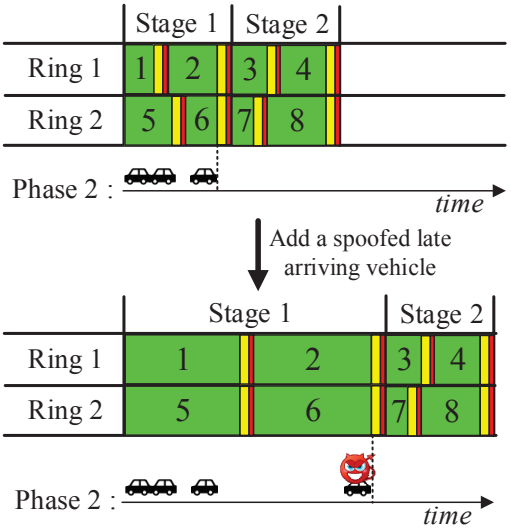

Fig. 6: Illustration of the last vehicle advantage. By exploiting it, even the spoofed data from a single attack vehicle can significantly influence the signal planning.

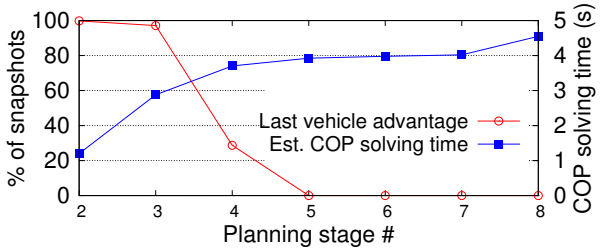

Fig. 7: Percentage of snapshots vulnerable to the last vehicle advantage and the estimated COP solving time with two to eight planning stages.

the infrastructure-side devices have more computation power. Thus, we are also interested in exploring whether the I-SIG system is still vulnerable after the last vehicle advantage is largely mitigated, i.e., with more planning stages configured. Thus, next we perform analysis for the I-SIG system with five-stage planning, with which exploiting last vehicle advantage is no longer the most successful trial (shown in Fig. 7).

2) Five-stage Planning Results: Column 3 in Table I shows the results after we configure the COP algorithm to use five-stage planning. As shown, even though the success rate is still high, the attack is much less effective: both the increased total delay time and percentage are nearly 7× less. Thus, without the last vehicle advantage, the I-SIG system becomes much less vulnerable to the data spoofing from one attack vehicle.

Nevertheless, the attacks can still cause a 11.5% total delay increase on average. Considering that the benefit of using the I-SIG system is around 26.2% total delay reduction [6], the attack result still shows moderate effectiveness. We analyze the causes and find two types of effective spoofing trials:

- Open a skipped phase. If there are skipped phases, the attacker can add the spoofed vehicle to one of them to force the signal planning to open it. Since an open phase needs at least $t_{g_{min}}$ green light time, which is 7 seconds in our generic intersection settings [37], this causes the signal plan under attack to waste the time in serving an empty phase at the cost of the vehicle delay in other phases. If only trying this category of data spoofing options, the total delay increase percentage is 8.9%, which is already very close to that (11.5%) with all data spoofing options enumerated.

- Extend the green light end time. Besides opening a skipped phase, the most successful data spoofing options are to set the spoofed vehicle arrival time to a few seconds after the original green light end time for a phase. This vehicle needs to wait for a whole planning stage if its serving is delayed to the next serving opportunity, which increases its delay and also the total delay by 20-50 seconds depending on the length of the next planning stage. Thus, in COP it is sometimes more cost effective by just extending the original green light end time for a few seconds to serve this vehicle. However, such extension is usually at most 4 seconds since it is no longer cost effective if the total delay added to the vehicles waiting in the subsequent phases is too much.

The data spoofing options for these two categories in total has around 10.1% in the total delay increase percentage. For the remaining 1.4% difference to that with all options enumerated, we find that the left-out successful trials are highly dependent on the traffic pattern and do not have a clear pattern.

C. The Transition Period

In this section, we analyze the vulnerability status of the I-SIG system in the transition period. The analysis results are shown in Column 4 to 15 in Table I. In the transition period, strategy S2 can now be used in addition to S1. Thus, in this section we analyze both strategies for each PR and planning stage configuration. As described in §V-B, S2 can add a number of non-existing unequipped vehicles by exploiting

the queue length estimation in the EVLS algorithm. Since there are around 100 vehicles in each snapshot, these non-existing vehicles constitute a substantial share of total vehicles in the signal planning. This should trick the COP algorithm into giving more priorities to this big group of non-existing vehicles at the cost of other vehicles' delay.

Overall effectiveness. As shown in the table, for a combination of a PR, a planning stage configuration, and a data spoofing strategy, the attack success rates and the average total delay increase percentages are 94.0% and 38.2% on average. This show that both strategy S1 and S2 are effective in creating congestion and can completely reverse the mobility benefit of using the I-SIG system. Also, we find that for each combination, the three attack effectiveness metrics are relatively the same, with less than 6% absolute differences in the average total delay increase percentages. This shows that the attack effectiveness is not significantly affected by PR. Next, we perform more in-depth analysis for the attacks on the two-stage and five-stage planning configurations respectively.

1) Two-stage Planning Results: Column 4-5, 8-9, and 12-13 in Table I shows the attack results for the two-stage planning. As shown, strategy S1 can still achieve over 99.1% success rate, and increase over 60.6% in the total delay. We find that the underlying cause is the same as that for the full deployment period: the last vehicle advantage (§VI-B1). Since the arrival time to maximally extend the green light time of the phases is not affected by the traffic conditions, the last vehicle advantage can always be reliably exploited for the two-stage planning scenarios regardless of the PR.

Strategy S2, which is newly enabled in the transition period, also shows high effectiveness. For all three PRs, the attack success rates are over 98.3%, and the average total delay increase percentages are over 32.4%. However, the increased percentages are still around 50% less than those using S1. We compare the most successful data spoofing options from S1 and S2, and find that for 99.0% of the snapshots, the best trial from S1 is no less than that from S2. We find that this is because even though adding the non-existing vehicles can indeed cause the signal planning to extend the t_g of a target phase to $t_{g_{max}}$ like S1, last vehicle advantage is able to further cause the t_g of the preceding phases to extend so that the vehicles in the subsequent phases can be further delayed.

2) Five-stage Planning Results: Column 6-7, 10-11, and 14-15 in Table I shows the results for five-stage planning. As shown, since the last vehicle advantage is much less effective for five-stage planning, the success rates and average delay increase percentages for S1 reduce to at most 83.2% and 17% respectively, as opposed to at least 99.1% and 60.2% for two-stage planning. Very similar to the full deployment period, we find that the most successful data spoofing trials are opening a skipped phase and extending green light end time.

Thus, with the last vehicle advantage becoming much less effective, S2 is now the dominating strategy. We compare the results between these two strategies for each snapshot, and find that for 93.5% of the snapshots, the best trial from S2 is no less than that from S1. We then analyze which data spoofing trials in S2 are the most successful. We find that for a certain phase, the best trial is to add the most non-existing unequipped vehicles, i.e., adding a farthest stopped vehicle using S2. If we only try these 8 options (one for each phase), the best trials among them and those among all possible data spoofing options only have 0.009% differences in the average total delay increase percentage. This is expected since adding more non-existing vehicles should gain more priority in signal planning and thus cost more delay to the other vehicles. For very few cases these 8 options fail to hit the most successful data spoofing trial. This is caused by the differences between the estimated and actual arrival time of the unequipped vehicles; if we calculate the attack effectiveness based on the estimated arrival time from the EVLS algorithm, these 8 options are always the best. Thus, in our exploit construction later, we only need to consider these 8 options, which is much less than trying all (usually over 250) possible options.

VII. Exploit Construction

Real-time attack requirement. In the last section, to understand the upper bound of the attack effectiveness, we enumerate all data spoofing options, which takes around 24.5 minutes on average on a single core computer. Since we only assume the attacker to have a consumer laptop that has four processors with usually around 3× speedup, this full enumeration takes 8 minutes on average. However, in practice the attack decision needs to be made fast enough so that the traffic condition does not change so much that the attack decision no longer applies.

Thus, to explore the end-to-end exploitability of the identified congestion creation vulnerabilities, in this section we take the real-time attack requirement into consideration and leverage the insights from our analysis in the last section to perform practical exploit construction.

A. Attack Decision Process

To meet the real-time attack requirement, our exploit construction uses a budget-based attack decision process. In this process, the attacker first passively tracks the phase changes. Once the phase in the current stage turns yellow, the attacker waits for 1 second and then triggers the decision process. This is based on our observation that after one second of yellow light all moving vehicles slow down and their trajectories start to stabilize. Since typically $t_y + t_r$ is 6 seconds [37], this gives the attacker up to 5 seconds of decision time.

In the decision period, the attacker first predicts the vehicle trajectory data at the next signal planning time. Like in the trajectory awareness component in the I-SIG system (§II-B2), the attacker maintains a vehicle trajectory database to store data like location, speed, and acceleration for the equipped vehicles based on the received broadcast BSM messages. In the prediction, the attacker assumes that the vehicles maintain their accelerations and thus predicts their speeds and locations after 5 seconds. In this step, the attacker needs to use the intersection map obtained from the reconnaissance step (§III) to determine whether a vehicle passes the stop bar of that lane after 5 seconds. If so and the current acceleration value is negative, we predict that it plans to have a hard stop at the stop bar and set the stop bar location as the predicted location.

Next, the attacker needs to make decisions about whether to attack, and if so, what data spoofing option to use. According to our vulnerability analysis, some of the most successful data

spoofing trials are related to the signal plan without attack, e.g., the green light end time. Thus, the attacker first runs the I-SIG system for the predicted vehicle trajectory data without trying any data spoofing option. Using the output signal plan and total vehicle delay without attack, the attacker then tries several data spoofing options just like in the vulnerability analysis, and pick the most successful one to use in the actual attack.

Since running the I-SIG system is time consuming, a trial budget is used to ensure that the whole decision process can finish in 5 seconds. Assuming the other parts, e.g., the BSM transition time and other local computation time, take less than 1 second (which typically take much less), we spare 4 seconds in total for (1) running the I-SIG system without attack, and (2) trying the data spoofing options. Since these trials are independent to each other, we use parallel computation to accelerate this part. We first measure the running time for the signal planning without attack, t_{normal}, and then calculate the trial budget as $3 \times \frac{4-t_{normal}}{t_{normal}}$, as the personal laptop with four processors in our lab is measured to have around $3\times$ speedup. With this, the attacker can plan their trials under this budget. The detailed budget-based trial strategies for different attack scenarios are described in the next section.

Based on the trial results, the attacker finds the data spoofing option with the highest total delay increases. If such increase is larger than zero, the attacker uses the corresponding data spoofing option to construct the BSM message and broadcast it out. Otherwise, the attacker does not attack.

B. Exploitation Strategy

In this section, we describe the exploitation strategies, i.e., the budget-based data spoofing trial strategies, for different combinations of PRs and planning stage configurations. The bottom half in Table VII summarizes the attack effectiveness for the constructed exploits in this section.

E1: Congestion attack for two-stage planning:

(1) In the first stage, if there are no skipped phases, try the data spoofing option with the latest arrival time for any of the two latter phases in stage 1, and then jump to (3). Trying the latter phases is because their latest vehicles are able to further extend the t_g of the two former phases to $t_{g_{max}}$.

(2) In the first stage, if there is a skipped phase, try the data spoofing option with the latest arrival time for this phase, and then jump to step (3). If there are two skipped and the budget allows more trials, try both and then jump to step (3). This is because opening an originally skipped phase can cause more total delay increase as explained in §VI-B2.

(3) In the second stage, if there are no skipped phases, try the two data spoofing options with the latest arrival time for the two former phases. If the budget allows more trials, try the latest arrival time for the two latter phases. Try the former phases first is because their latest vehicles can cause phase sequence switches to further increase the delay.

(4) In the second stage, if there is a skipped phase, try the data spoofing options with the latest arrival time for this phase. If the budget allows more trials, try the latest arrival time for the former phases, and then the latter phases. If there are two skipped phase, try the two data spoofing options with the latest arrival time for these two phases.

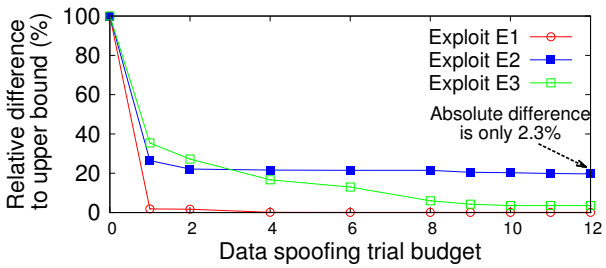

Fig. 8: Relative differences between the average delay increase percentages using the three exploits with limited trial budgets and those by trying all possible options.

As introduced in §II-B2, at each planning time only the planned duration for the first stage is immediately applied. Thus, in the above strategy we prioritize the attacking on the first stage so that the attack has an immediate effect. Also, in this strategy we only consider at most two skipped phase since we do not observe any snapshot in our analysis has more than two skipped phases under the normal traffic demand.

E2: Congestion attack for five-stage planning in the full deployment period:

(1) If there are skipped phases, try any data spoofing option for each of these phases. If the budget is not enough, prioritize the ones in the earlier stages.

(2) Try the data spoofing options b_g seconds after the originally green end time for each open phase. For the first time entering this step, b_g is 1. If the budget is not enough, prioritize the ones in the earlier stages.

(3) If the budget allows more trials, repeatedly try (2) with b_g being increased by 1 each time until the budget is used up.

E3: Congestion attack for five-stage planning in the transition period:

(1) For the through phases, try the data spoofing options that add Q_p non-existing queuing unequipped vehicles for each phase p. If the budget is not enough, prioritize the ones in the first stage.

(2) For the left-turn phases, try the data spoofing options that add Q_p non-existing queuing unequipped vehicles for each phase p. If the budget is not enough, prioritize the ones in the first stage.

(3) If the budget allows more trials, repeatedly try (1) and (2) with Q_p being decreased by 1 each time until the budget is used up.

In this strategy, we prioritize the through phases since their lanes are longer than those of the left-turn phases, and thus has much (usually twice) larger Q_p.

Fig. 8 shows the attack effectiveness of these three exploits with different trial budget on the snapshots in the vulnerability analysis. In the figure, the attack effectiveness metric is the average total delay increase percentage. As shown, for E1, only 4 trials are need to reach the upper bound attack effectiveness, i.e., the one by trying all possible options. For E2, the attack effectiveness converges quickly after using 2 trials, and then decreases very slowly when b_g increases with more available budget. At the tail, the relative difference to the upper bound

attack effectiveness is around 20%, but since the upper bound is only 11.5%, it only has 2.3% absolute difference. As discussed in §VI-B2, the best trials responsible for such difference highly depend on specific traffic patterns. For E3, only 8 trials are need to reach the upper bound attack effectiveness, which is consistent with the discussion in §VI-C2.

We implement this budget-based trial strategies, and evaluate their effectiveness on the snapshots in the vulnerability analysis. In this experiment, we use the running time without attack for each snapshot to dynamically choose trial budget. The results is shown at the bottom half in Table VII. As shown, even though two-stage planning is much faster than five-stage planning, the maximum trial number needed for E1 is only 6 so the average trial number is 3.6-3.8. For five-stage planning scenarios, in the worst case the attacker can at most try 13.3 options due to the real-time attack requirement. This is already much less than trying all possible options, which needs nearly 1000 trials for S1 and around 250 options for S2. Nevertheless, our trial strategies show high effectiveness with less than 2.2% difference to the upper bound attack effectiveness.

VIII. Attack Evaluation

In this section, we implement and evaluate the constructed exploits using simulations with real-world intersection settings.

A. Evaluation Setup

Real-world intersection settings. In this evaluation we use the map of a real-world intersection with its real phase configurations. The intersection map is shown in the screenshot in Fig. 1. Compared to the generic intersection structure, this intersection has different speed limits on each approach. The speed limits are 30 mph, 35 mph, 40 mph, and 45 mph for southbound, eastbound, northbound and westbound respectively. Only northbound approach has dedicated right turn lane, while in other approaches the right turn lane is shared with the through lane. The map range of the eastbound approach is only extended to 220 meters because of the existence of a close-spaced upstream intersection.

Real-world traffic demand. To increase the practicality of our analysis, we use the real traffic demand for this intersection in our VISSIM configuration. To measure such demand, we went to the intersection and videotaped the traffic in the intersection on May 16th, 2017, 4-5 pm. Based on the videos, we manually counted the passing vehicles for each lane, and calculated the traffic demand of each approach and the turning ratio for each lane (the possibility of turning left or right for the vehicles), as the input to the VISSIM traffic model.

Experiment setup. In the experiment, the I-SIG system and attack program can receive the BSM messages within their DSRC ranges. The DSRC ranges for all approaches are set to the normal value, 300 meters, except the one for the eastbound approach is 220 meters as its lanes are shorter. On the attacker side, the BSM messages are used in the attack decision process detailed in §VII-A. After that, the BSM message sent with the spoofed data is merged with the other BSM messages. The I-SIG system uses these BSM messages, which may or may not have the attack message, to perform the signal planning and then use the plan to control the traffic signals in VISSIM.

For each combination of PR and planning stage configuration, we run the experiments for one hour three times, each with a different random seed, based on the aforementioned real-world traffic demand. In this experiment we launch the attack continuously for every signal planning in the I-SIG system. This is different to the experiments in the vulnerability analysis in which the attacks are launched individually to each snapshot. In comparison, such continuous attacking is closer to real-world attack situations. As we will show later, this is able to create a cumulative attack effect and thus create even more congestion than that in the vulnerability analysis.

Attack effectiveness measurement. In the evaluation we directly measure the vehicle travel delay using each vehicle's trajectory output by VISSIM. To calculate the per-vehicle delay, we subtract the free-flow travel time, i.e., the travel time at the speed limit, from the vehicle's actual travel time. Then the total vehicle delay is calculated as the sum of the per-vehicle delay for all vehicles generated in the experiment. In the VISSIM simulation, for the same random seed the vehicle generated with the same ID has exactly the same initial data, e.g., the same generation time and the same initial speed and location. Thus, both the total vehicle delay and the per-vehicle delay for experiments with and without attack are comparable.

B. Results

The results are summarized in Table II and analyzed below:

1) Exploit E1 and E2: Column 2, 4, 6, and 8 show the results for E1. As shown, E1 is able to increase more than 60% of the total delay for all cases expect when the PR is 25%. These results are consistent with those in Table I, showing high attack effectiveness. When the PR is 25%, we find that the errors in the unequipped vehicle estimations in the EVLS algorithm are greater than those in the generic intersection settings, causing the attack effectiveness to decrease. Nevertheless, the total delay increase percentage is still very high (46.2%): for a vehicle, a one-hour trip now takes nearly one and half hours, showing a significant decrease of the transportation mobility.

The results for E2 are shown in Column 3. As shown, the attack effectiveness is only 4.8%, which is around 50% lower than that in the vulnerability analysis. We find that this is because both categories of the successful data spoofing trials in §VI-B2 can be largely affected by errors in the vehicle trajectory data prediction in our attack decision process (§VII-A). For the one that opens the skipped phase, any legitimate vehicle requesting that phase in 5 seconds nullifies the attack effect. For the one that extends the green light end time, the original green light end time can vary after 5 seconds due to changes in the arrival table. Among the three exploits, E2 is the most dependent on traffic conditions and thus more sensitive to the errors in our prediction. Considering that it also has the least attack effectiveness, E2 is thus the least attractive exploit among the three.

2) Exploit E3: The results for E3 are shown in Column 5, 7, and 9. Surprisingly, we find that these attacks are much more effective than those in the vulnerability analysis: when the PRs are 75% and 50%, the average delay increase percentages are 181.6% and 193.3%, which are over 5× more than those in the vulnerability analysis. The increase for the 25% PR scenario is a bit lower, but is still around 4× more.

CV deployment	Full deployment		Transition period					
	100% PR		75% PR		50% PR		25% PR	
COP config.	2-S	5-S	2-S	5-S	2-S	5-S	2-S	5-S
Exploit	E1	E2	E1	E3	E1	E3	E1	E3
Ave. delay inc. (s) & %	68435.4	4695.9	64008.0	187746.0	66797.4	197410.0	56618.0	146685.0
	66.7%	4.8%	61.7%	181.6%	64.2%	193.3%	46.2%	133.2%

TABLE II: Evaluation results for the practical exploits. PR is short for penetration rate. Two-stage planning and five-stage planning in COP are denoted as 2-S and 5-S, with the former being the default choice.

The lane blocking effect. We find that such significant increase is because continuous attacking is able to cause the attack effect to accumulate, and thus greatly escalates the attack effectiveness. More specifically, in five-stage planning, since the planning is allowed to delay serving some vehicles in the current stage for more optimal long-term benefit, these vehicles are attacked for another time in the next signal planning time. If the vehicle is near the end of the queue, it can be attacked multiple times. Since in the vulnerability analysis we only estimate the effectiveness for attacking once, such cumulative attack effect causes the average total delay to significantly increase in comparison to that in the vulnerability analysis. Such cumulative attack effect does not exhibit for the two-stage planning scenarios, since two-stage planning only has one serving opportunity for each phase and it is not allowed to delay serving any vehicle.

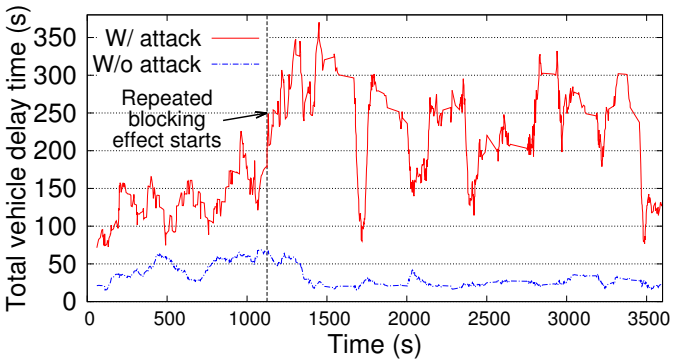

Fig. 9: Average vehicle delay every one minute with and without attack. The repeated blocking effects start at around second 1125.

We further find that such cumulative attack effect is able to cause an even higher level of congestion, which *can block an entire approach, causing massive traffic jams*. This is because with such effect the queues in the left-turn lanes cannot be effectively released and thus begin to increase with time. Since the left-turn lanes are shorter in nature, at a certain point the queues start to spill over to the through lanes and block the through lane. This causes the through lane to start queuing after the spilled-over left-turn vehicles. With both the real queuing vehicles and the non-existing unequipped vehicles added by our attack in the through lanes, the COP algorithm sees more than 80 vehicles queuing in the through lanes and thus only gives the spilled-over left-turn phase the minimum green light time. Thus, the left-turn phase can now only release the fewest possible vehicles. When some spilled-over vehicles finally enter the left-turn lane, the following left-turn vehicles quickly block the through lanes again.

Such blocking effect is shown earlier in Fig. 1, which is a screenshot taken at the 1785.80 second in the VISSIM simulation for one of the three random seeds and the 75% PR. Note that such spillover and blocking effect always appears on at least one approach in all E3 experiments. As shown in the figure, in both the northbound and southbound approaches, the left-turn vehicles spill over and block the through lanes, causing long queues in the approach. In the real-world traffic demand we collected from 4 to 5 pm, the northbound approach has the most left-turn vehicles and thus is the earliest to block and thus have the longest queue at the time of the screenshot.

Fig. 9 shows the average delay every one minute with and without attack in the northbound approach in this experiment. As shown, the delay under attack usually has an increase when the delay without attack increases. This is because when the approach is more congested without attack due to a temporarily higher demand, the congestion attack can further escalate such congestion. As shown, at around second 1125, such higher demand is leveraged to create the blocking effect, and thus the congestion level is significantly increased. After 10 minutes, the spillover is finally cleared, but in as short as 1 minute, the blocking effect happens again. In the figure, we can see such repeated blocking effect till the end of the experiments. In the traffic jam period starting from second 1125 till the end, nearly 600 vehicles arrive and around 50% of them need to spend nearly three minutes for an originally half-minute trip (27.7 seconds on average), and around 22% need to spend over 7 minutes, which is 14 times higher. This means that for these 22% of vehicle, if the trip involves a series of intersections, i.e., in a corridor, a 10-minute trip can now cost over 2 hours.

IX. Defense Discussions

As shown in our study, even though the I-SIG system has shown high effectiveness in benign settings, the current algorithm design and configuration choices are highly vulnerable to data spoofing. To proactively addressed these problems before larger-scale deployment, this section discusses defense directions based on the insights from our analysis.

Robust algorithm design for the transition period. As concretely shown in our evaluation, the most effective congestion attack is on the transition period: the total delay increase percentage is nearly 200%, and by continuously attacking for less than 20 minutes, it is able to trigger the blocking effect on an entire approach, causing massive traffic jams. According to the current I-SIG system design, such problem can only be largely alleviated when PR reaches more than 95%. This is thus the most urgent problem in the current I-SIG system design: the market penetration rate of CV technology needs to start somewhere, and thus it inevitably needs to go through a transition period. Even after all new light-duty vehicles are mandated to install OBUs, which is exactly what the USDOT is proposing now, there are still heavy duty vehicles and old vehicles on the roads. As estimated by the USDOT [16], it may

take 25-30 years to reach a 95% PR after it starts such mandate. Thus, if such system cannot handle the security challenges for the transition period, it is not robust enough to get the larger-scale deployment even started in practice.

Fundamentally, this is caused by the lack of a sufficiently robust signal control algorithm for the transition period. As introduced in §II-B2, the COP algorithm is a suitable design choice for the CV-based signal control, but it is only optimal in the full deployment period. To ensure that the I-SIG system can still be effective when PR is low, the current design tries to infer the unequipped vehicle data to solve the dilemma. However, if such inference is not robust, it can be greatly manipulated for malicious purposes — which is exactly what we have uncovered in this study. Since the amount of vehicle data input is much less than that in the full deployment period, any signal control algorithm for the transition period is inherently more sensitive to data spoofing attacks, making it fundamentally more challenging to ensure the robustness. Considering that the transition period is unavoidable and may last as long as 30 years, we believe that this calls for a joint research effort among both the transportation and the security communities to design effective and robust signal control algorithms specifically for the transition period.

Performance improvement for RSUs. As introduced in §II-B2 and analyzed in §VI, the arrival time based signal planning in the COP algorithm is very suitable for the CV-based signal control, and when given enough computation power, such planning is indeed very hard to be maliciously influenced by small amounts of spoofed data in the full deployment period. Unfortunately, due to the limited performance in today's RSUs, the I-SIG system has to use a suboptimal implementation of the COP algorithm, which is found to introduce the last vehicle advantage, allowing the data from one single attack vehicle to significantly influence the signal control. Because of this, even if the security challenge for the transition period is addressed, the I-SIG system can still be greatly manipulated by data spoofing attacks. Thus, it is important to improve the performance of today's RSUs so that more optimal configurations can be used in the traffic control. Such improvement can be at both the software level, e.g., code optimization, and the hardware level, e.g., CPU and memory upgrades. Such performance improvement is generally beneficial since more computation capabilities can help better balance the trade-off between security and performance.

Data spoofing detection using infrastructure-controlled sensors. Besides improving the robustness at the control algorithm level, another defense direction is to detect and filter the BSM messages with spoofed data on the infrastructure side. Since these messages are still correctly signed, such defense must rely on data validity checks. Unfortunately, in the current design, the I-SIG system only has one data source about the attack vehicle — the attacker-controlled trajectory data via BSM messages [6]. Thus, any data validity check methods based on this are unlikely to be effective since the attacker can strategically control the spoofed data so that the vehicle trajectories appears perfectly normal.

Thus, to ensure high effectiveness, data spoofing detection on the infrastructure side needs to rely on data sources that attackers cannot easily control, e.g., infrastructure-controlled sensors, to cross validate the data in BSM messages. We find that there are actually existing infrastructure-side sensors ready to be used for this purpose. For example, the vehicle detectors buries underneath the stop bar of each lane was used to measure aggregated traffic information in today's traffic control. Even though they are less useful in the CV environment, they may be re-purposed to help detect data spoofing, which may be a cost effective solution since they are installed already. If such aggregated data is not sufficient, the infrastructure side may need to install sensors with more informative data, e.g., cameras. One challenge in this direction is how to best leverage different types of infrastructure-side sensors to design a detection system that is both accurate and hard to evade, which we leave as future work.

X. Related Work

Data spoofing attack in the CV environment. Similar to our work, previous work also identifies data spoofing as a realistic attack vector in the CV environment. Amoozadeh et al. studied the V2V-based automated vehicle platoon system, and found that spoofed attacks can cause rear-end collision or significant instability [17]. A more recent work summarizes a comprehensive list of data spoofing attack sources including not only DSRC but also other sensors such as GPS [23]. While these work focus on data spoofing attacks on V2V, our paper is the first study that exposes concrete data spoofing attacks on the transportation infrastructure side through V2I. Compared to V2V attacks that can at most affect one lane of vehicles at a time, V2I attacks can affect all vehicles in an intersection as concretely shown in our evaluation, and thus are able to cause much wider impact on the transportation system.

Critical infrastructure security. Several studies have investigated the security of critical infrastructure and facilities, e.g., smart grid [18], [31]. These studies highlight the security challenges and the severe consequences brought by introducing connectivity into these previously isolated critical systems, which is also concretely shown in this work for the next-generation CV-based transportation. Closer to our work, Ghena et al. performed the first publicly available security analysis of a deployed traffic infrastructure system [27]. Their work found that the traffic controllers uses weak credentials and can be remotely controlled by the attacker. In comparison, our work targets the next-generation CV-based traffic control instead of the traditional one. In addition, the weak credential problem they discovered is a known problem across many embedded network devices [22], and can be fixed using state-of-the-art authentication mechanisms [27]. In comparison, our study assumes that such problem has already been solves, and targets new security problems at the traffic control algorithm level.

Traffic control algorithm security. Prior to our study, very few studies explored the security problems in the traffic control algorithms. Laszka et al. performed a theoretical analysis to estimate the potential congestion an attacker can create assuming that she can arbitrarily compromise multiple signal controllers [30]. A follow-up study was then performed for the same attack goal but with a weak assumption, in which the attacker can only compromise the sensors that collects traffic flow information [26]. In comparison, neither of these work analyzes the CV-based signal control scenario targeted in our work. In addition, compared to their thread model that assumes the ability of compromising arbitrary numbers

of infrastructure-side devices, our threat model, data spoofing from one signal attack vehicle, is much more realistic (§III).

XI. Concluding Remarks

In this work, we perform the first security analysis of the emerging CV-based signal control system. Targeting a highly realistic threat model, data spoofing from one single attack vehicle, we perform vulnerability analysis and find that the current signal control algorithm design and configuration choices are highly vulnerable to congestion attacks. The evaluation results under real-world settings validate the attack effectiveness and show that the attacks can even create a blocking effect that jams whole approaches. Defense directions are then discussed leveraging the insights.

This work serves as a first step to understand the new security problems and challenges in the next-generation CV-based transportation systems. It is expected to inspire a series of follow-up studies, including but not limited to (1) more extensive evaluation with different intersection sizes and traffic patterns, (2) more extensive analysis considering other CV-based transportation systems, algorithms, and security implications, (3) more concrete defense system design and evaluation.

Acknowledgments

We would like to thank Andre Weimerskirch, Yuru Shao, and the anonymous reviewers for providing valuable feedback on our work. This research was supported in part by an award from Mcity at University of Michigan, the National Science Foundation under grants CNS-1318306 and CNS-1526455 and by ONR grant N00014-14-1-0440.

References

[1] "Cohda Wireless OBU and RSU," http://cohdawireless.com/Products/Hardware.aspx.

[2] "Connected Vehicle Applications," https://www.its.dot.gov/pilots/cv_pilot_apps.htm.

[3] "Connected Vehicle Pilot Deployment Program Phase 1, Concept of Operations (ConOps) Tampa (THEA)," https://ntl.bts.gov/lib/57000/57000/57032/FHWA-JPO-16-311.pdf.

[4] "CV application: MMITSS-AZ 1.0," https://www.itsforge.net/index.php/community/explore-applications/for-search-results#/30/63.

[5] "DSRC: The Future of Safer Driving," https://www.its.dot.gov/factsheets/dsrc_factsheet.htm.

[6] "MMITSS Final ConOps: Concept of Operations," http://www.cts.virginia.edu/wp-content/uploads/2014/05/Task2.3._CONOPS_6_Final_Revised.pdf.

[7] "Open Source Application Development Portal (OSADP)," https://itsforge.net/.

[8] "PTV Vissim," http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim.

[9] "Savari StreetWAVE RSU," http://savari.net/technology/road-side-unit.

[10] "Traffic jams cost U.S. drivers $1,200 a year: study," http://www.reuters.com/article/us-autos-congestion-idUSKBN15Z0DL.

[11] "US Department of Transportation hopes to mandate V2V communications," https://www.cnet.com/roadshow/news/us-department-of-transportation-hopes-to-mandate-v2v-communications.

[12] "U.S. DoT Connected Vehicle Pilot Deployment Program," https://www.its.dot.gov/pilots/.

[13] "USDOT: 20 Questions About Connected Vehicles," https://www.its.dot.gov/cv_basics/cv_basics_20qs.htm.

[14] "USDOT: Multi-Modal Intelligent Traffic Safety System (MMITSS)," https://www.its.dot.gov/research_archives/dma/bundle/mmitss_plan.htm.

[15] "USDOT: Security Credential Management System (SCMS)," https://www.its.dot.gov/factsheets/pdf/CV_SCMS.pdf.

[16] "Vehicle-Infrastructure Integration (VII) Initiative: Benefit-Cost Analysis," https://www.pcb.its.dot.gov/connected_vehicle/508/Library/Library-RRs-Institutional/VII%20BCA%20Report%20Ver2-3.htm.

[17] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security Vulnerabilities of Connected Vehicle Streams and Their Impact on Cooperative Driving," in *IEEE Communications Magazine*, 2015.

[18] R. Anderson and S. Fuloria, "Who Controls the Off Switch?" in *IEEE Smart Grid Communications (SmartGridComm)*, 2010.

[19] W. Burghout and J. Wahlstedt, "Hybrid Traffic Simulation With Adaptive Signal Control," in *Transportation Research Record: Journal of the Transportation Research Board*, 2007.

[20] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner *et al.*, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," in *USENIX Security*, 2011.

[21] K.-T. Cho and K. G. Shin, "Fingerprinting Electronic Control Units for Vehicle Intrusion Detection," in *USENIX Security Symposium*, 2016.

[22] A. Cui and S. J. Stolfo, "A Quantitative Analysis of the Insecurity of Embedded Network Devices: Results of a Wide-area Scan," in *ACM ACSAC*, 2010.

[23] D. Dominic, S. Chhawri, R. M. Eustice, D. Ma, and A. Weimerskirch, "Risk Assessment for Cooperative Automated Driving," in *ACM CPS-SP Workshop*, 2016.

[24] M. Emmelmann, B. Bochow, and C. Kellum, "Vehicular Networking: Automotive Applications and Beyond." John Wiley & Son, 2010.

[25] Y. Feng, K. L. Head, S. Khoshmagham, and M. Zamanipour, "A Real-time Adaptive Signal Control In A Connected Vehicle Environment," 2015.

[26] A. Ghafouri, W. Abbas, Y. Vorobeychik, and X. Koutsoukos, "Vulnerability of Fixed-time Control of Signalized Intersections to Cyber-tampering," in *IEEE Resilience Week (RWS), 2016*, 2016.

[27] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, and J. A. Halderman, "Green Lights Forever: Analyzing the Security of Traffic Infrastructure," in *Usenix WOOT*, 2014.

[28] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," 2011.

[29] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental Security Analysis of a Modern Automobile," in *IEEE S&P*, 2010.

[30] A. Laszka, B. Potteiger, Y. Vorobeychik, S. Amin, and X. Koutsoukos, "Vulnerability of Transportation Networks to Traffic-signal Tampering," in *ACM ICCPS*, 2016.

[31] J. Liu, Y. Xiao, S. Li, W. Liang, and C. P. Chen, "Cyber Security and Privacy Issues in Smart Grids," in *IEEE Communications Surveys & Tutorials*, 2012.

[32] C. Lozoya, P. Martí, M. Velasco, J. M. Fuertes, and E. X. Martin, "Resource and Performance Trade-offs in Real-time Embedded Control Systems," in *Real-Time Systems*, 2013.

[33] H. C. Manual, "Highway Capacity Manual," in *Transportation Research Board*, 2000.

[34] S. Mazloom, M. Rezaeirad, A. Hunter, and D. McCoy, "A Security Analysis of an In-Vehicle Infotainment and App Platform," in *Usenix WOOT*, 2016.

[35] V. Narayanan and Y. Xie, "Reliability Concerns in Embedded System Designs," in *IEEE Computer*, 2006.

[36] S. Sen and K. L. Head, "Controlled Optimization of Phases at an Intersection," in *Transportation science*, 1997.

[37] T. Urbanik, A. Tanaka, B. Lozner, E. Lindstrom, K. Lee, S. Quayle, S. Beaird, S. Tsoi, P. Ryus, D. Gettman *et al.*, "Signal Timing Manual," in *Transportation Research Board*, 2015.

[38] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn, "A Security Credential Management System for V2V Communications," in *IEEE Vehicular Networking Conference (VNC)*, 2013.