Application level attacks on Connected Vehicle Protocols

Ahmed Abdo
Department of Electrical and Computer Engineering University of California, Riverside
Email: aabdo003@ucr.edu

Sakib Md Bin Malek
Department of Computer Science and Engineering University of California, Riverside
Email: sbin003@ucr.edu

Zhiyun Qian
Department of Computer Science and Engineering University of California, Riverside
Email: zhiyunq@cs.ucr.edu

Qi Zhu
Department of Electrical and Computer Engineering Northwestern University
Email: qzhu@northwestern.edu

Matthew Barth
Department of Electrical and Computer Engineering University of California, Riverside
Email: barth@ee.ucr.edu

Nael Abu-Ghazaleh
Department of Computer Science and Engineering University of California, Riverside
Email: naelag@ucr.edu

Abstract

Connected vehicles (CV) applications are an emerging new technology that promises to revolutionize transportation systems. CV applications can improve safety, efficiency, and capacity of transportation systems while reducing their environmental footprints. A large number of CV applications have been proposed towards these goals, with the US Department of Transportation (US DOT) recently initiating three deployment sites. Unfortunately, the security of these protocols has not been considered carefully, and due to the fact that they affect the control of vehicles, vulnerabilities can lead to breakdowns in safety (causing accidents), performance (causing congestion and reducing capacity), or fairness (vehicles cheating the intersection management system). In this paper, we perform a detailed analysis of a recently published CV-based application protocol, Cooperative Adaptive Cruise Control (CACC), and use this analysis to classify the types of vulnerabilities that occur in the context of connected Cyber-physical systems such as CV. We show using simulations that these attacks can be extremely dangerous: we illustrate attacks that cause crashes or stall emergency vehicles. We also carry out a more systematic analysis of the impact of the attacks showing that even an individual attacker can have substantial effects on traffic flow and safety even in the presence of message security standard developed by US DOT. We believe that these attacks can be carried over to other CV applications if they are not carefully designed. The paper also explores a defense framework to mitigate these classes of vulnerabilities in CV applications.

1 Introduction

The United States Department of Transportation (US DOT) has been developing next-generation Intelligent Transportation Systems (ITS) [2] where vehicles and transportation infrastructure communicate and collaborate towards goals such as improving safety, increasing traffic flow capacity, supporting driver assistance functionality, and reducing overall carbon footprint [16]. Some of these technologies are already installed across the country such traffic signal coordination, transit signal priority, and traveler information systems.

One widely deployed early example of such functionality is Intelligent Traffic Signal Systems (I-SIG), which have been deployed in several cities, reducing the average traffic delay by 26.6% [24]. While I-SIG involves only making the infrastructure intelligent, another class of ITS applications involves vehicles communicating to coordinate with other vehicles and the infrastructure intelligently. The subset of ITS applications that involves vehicles communicating to each other (V2V) and the Infrastructure (V2I) are called *Connected Vehicles* (CV) applications. Many of the CV applications are starting to be prototyped and have reference implementations [25]. The US Department of Transportation (US DOT) has started testing applications in three deployment sites. Other experimental projects incorporating platooning are starting to emerge: e.g., a consortium of companies, universities and the Flemish government are building a test bed to experimentally test automated CV driving [1]. Tesla is also working on self-driving electric trucks that can move in platoons behind a designated lead vehicle [6].

In these initial stages where researchers and engineers are developing early prototypes of CV applications, security is not being considered deeply. CVs expose a large attack surface as an open systems with many participants and complex functionality: attacks may target application protocols, networking, sensing and vehicle control, with the potential to cause accidents, traffic delays and other harm to the system. A message security standard, the Secure Certificate Manage-

ment System (SCMS), has been defined by USDOT but it only ensures that cars and road side units have certificates that enable them to participate in communication [20].

Vulnerability and Attack Analysis: It is essential to understand the threats faced by CV protocols to understand how to design them securely. Towards this goal, this paper explores the vulnerabilities that arise at the application level of CV applications. We show that even when an attacker does not spoof or modify messages, it does not stop a malicious actor from obtaining a certificate, or a compromised participant with a valid certificate, from using it to falsify information in its messages. We present the threat model in Section 2. We conduct this analysis in the context of an important CV application called Cooperative Adaptive Cruise Control (CACC). CACC is used to group nearby cars into a platoon and adaptively control their speed. The vehicles in a platoon are subjected to reduced air drag as well as improvements in overall traffic flow, driving safety, capacity, and fuel economy. Section 3 introduces CACC. The application logic is complex, having to consider cases such as cars joining and leaving a platoon, merging and splitting of platoons, lane changes, and platoon leaders leaving. These maneuvers are triggered and coordinated through messages. An attacker can exploit this protocol by sending messages with false information leading to a number of possible attacks that reduce the safety, and performance of the system. In Section 4 we introduce five general classes of vulnerabilities that we believe that can be applied to networked cyber physical systems. We describe specific attacks against CACC in Section 5, showing a number of successful attacks even in the presence of SCMS.

Attack Demonstration and Evaluation: As CV systems are not deployed and/or generally available for public experimentation, to evaluate these attacks, we use a previously developed implementation of CACC in a state of the art vehicular simulator, VENTOS [9], that is widely used by practitioners and developers . We show scenarios where the vulnerabilities can be exploited to cause safety breakdowns or to interfere with an emergency vehicle. We define metrics for evaluating the attack impact that measures mobility (traffic throughput) and safety (average separation between cars). We show that attacks can substantially interfere with the operation of CACC leading to increased vehicular speeds and reduced safety margins. We present our results in Section 6.

Potential Mitigation: Having established these attacks on the CACC application level, we need to consider a mitigation framework in Section 7. We use the classification of the five vulnerability types we introduce to guide the design of the mitigation steps that either eliminate or interfere with them. We show that the defense indeed mitigates the vulnerabilities we identified in CACC without substantially harming performance.

2 Threat Model

We assume a CV application using Security Credentials Management System (SCMS) [7]. SCMS became available to coincide with the full-scale deployment of devices at three US DOT CV pilot sites (New York, Tampa, and Wyoming) [10–12]. The current implementation is a proof-of-concept Certificate-Based Authentication system that uses a Public Key Infrastructure [20] for certificate management. Pseudonym Certificates (PCs) are used and rotated to enable message authentication and validation without exposing the privacy of a vehicle by having a permanent certificate. A vehicle can enroll in the system by submitting an enrollment request to US DOT. PC can be obtained by vehicles for a short term, ranging from 5 minutes to few days, and is used for basic safety message (BSM) authentication. On Board Equipment (OBE) uses identification certificates to authenticate itself in V2I applications. However, none of the V2I applications we reviewed require encryption by the OBE at the application level.

SCMS prevents an attacker from falsifying messages from another vehicle as each message gets signed with a certificate. However, SCMS can not prevent a malicious actor from obtaining a certificate and participating in the protocol through replaying the messages while they are valid, or sending its own message, with fabricated data, using its certificate. Although it is currently unclear how well SCMS can function since it is not open source, we assume that it introduces no significant latency. In general, we do not consider message delays, jamming, physical attacks on sensors or controllers, DoS attacks, or any similar attacks to be part of our threat model since our focus is on application level exploitation. It is clear that such attacks are possible, and perhaps can be used in conjunction with application level attacks to amplify their damage. We also do not consider attacks exploiting bugs in the software stack of any of the existing components running on the infrastructure components, or other cars which we consider to be orthogonal to our threat model. We also do not consider physical attacks on the sensors of the vehicles or any sensors deployed by the infrastructure.

In some attacks, we assume that the attacker is a compromised vehicle which uses a radio that is capable of reaching cars farther away than typical vehicular radios and is capable of authenticating itself to the SCMS as a regular vehicle, then applying its attacks in the application level. We assume that the attacker knows the application logic and crafts its actions to manipulate this logic.

3 Cooperative Adaptive Cruise Control

In this section, we introduce the Cooperative Adaptive Cruise Control (CACC) application to provide background necessary to understand its potential security vulnerabilities. In CACC, a group of vehicles, with a close spacing between

them, can form a platoon if they are traveling in the same direction. Once created, vehicles in the platoon co-operate to travel at the same speed and make decisions as a group, maintaining reduced clearance gaps between each other, allowing for more efficient use of the highway and reducing the air drag compared to vehicles traveling individually. A Platoon Management Protocol (PMP) controls platoon operations and maneuvers. The leading/front vehicle acts as the coordinator and controls platoon decisions such as the speed, lane changes, and merging with other platoons. Vehicles communicate typically through Dedicated Short Range Communication (DSRC/IEEE 802.11p [21]), although eventually they may use 5G instead [22]. Road Side Units (RSUs) [19] are infrastructure units that are used to coordinate behavior or maneuver across cars, or to maintain shared certain state. Each vehicle has On Board Unit (OBU) that can uses Basic Safety Messages (BSMs) to send some periodic information such as speed and location and receive event messages such as those informing of traffic conditions in an area they are entering.

In our experiments we use PMP, which was proposed and developed by Amoozadeh et al [15]. PMP supports a number of maneuvers representing different operations that platoons could potentially perform. This section introduces some of the primary maneuvers.

Joining a new Platoon (or forming a new platoon): If a vehicle receives a beacon message sent from a vehicle ahead of it, it will evaluate the position, speed, acceleration, and other relevant information to determine whether or not to join the platoon. Beacon messages also contain a Platoon Id, which is a locally distinct number used to distinguished the various platoons in the area.

Split Maneuver: Split maneuver is always initiated by the platoon leader. When the platoon size exceeds the optimal platoon size, the maneuver can be used to break the platoon into two, at a specific position. First, a SPLIT_REQ message is sent to the vehicle where the split should occur. If the request is accepted, a SPLIT_ACCEPT message is sent back to the leader. Subsequently, the leader sends a unicast CHANGE_PL to the potential leader of the new platoon resulting from the split. Finally, the original leader will report split end by sending SPLIT_DONE message.

Merge Maneuver: In this maneuver, two platoons, traveling in the same lane and close to each other, merge to form one platoon. If the leader of the rear platoon discovers another platoon in front of it with capacity to merge, the leader sends a unicast MERGE_REQ to the front platoon leader. Once the front leader accepts the merge request, it sends back a MERGE_ACCEPT message. On receiving this message, the rear platoon leader starts a catch-up maneuver. Upon reaching the front platoon, the rear platoon leader sends CHANGE_PL to all its followers to change the platoon leader to the front leader. Now the followers start listening to the front leader and eventually the rear leader changes its state from leader to follower after sending a MERGE_DONE message.

	Vulnerability	Explanation
V1	Fake message contents	Attacker sends messages with false information
V2	Insufficient information	Critical data not communicated
V3	Inadequate identifier binding	Incorrect binding of physical object to logical object
V4	Incomplete or unsafe protocol logic	Protocol does not consider all scenarios
V5	Trust delegation	Decisions delegated to possibly malicious participant

Table 1: Vulnerability Classification in Networked Cyber-physical Systems

Leave Maneuver: The departing vehicle initiates the process by sending a LEAVE_REQ message. The leader sends a LEAVE_ACCEPT message and then split process starts. Once the leaving vehicle changes lane, a GAP_CREATED message is broadcast. A merge process begins to reduce the gap until the platoon has the target gap distance between each car.

Change Lane Maneuver: In this maneuver, the platoon leader decides that the platoon needs to change lane. A platoon might need to change lanes if the platoon need to exit the highway or if it has been given instruction from the RSU due to lane congestion. The platoon leader sends CHANGE_LANE instruction to all the other vehicles in the platoon and they perform the maneuver together following the leader's lane change. After that, all the followers send an ACK message to the leader, if they changed the lane successfully.

4 Vulnerability Analysis and Classification

It is tempting to consider networked cyber-physical systems such as CV as simply another networked system from the perspective of security, and indeed this is the case with respect to the vulnerability vectors. However, these systems differ in two important aspects with profound implications on vulnerabilities and defenses. The systems are (1) cooperative: they coordinate to accomplish a combined outcome; and (2) constrained by physics: protocol logic, as well as misbehavior outcomes are defined with respect to their impact on the system in the physical world, for example, considering both space and time.

The factors, outlined above, lead to vulnerability classes that are tied to the protocol logic and the physical system. Based on our analysis of multiple CV applications, we identified a number of vulnerability classes, which we believe generalize to other networked cyber-physical systems as well. These vulnerabilities arise even if vehicles have a certificate, which, to begin with, is not that difficult to obtain.

The first vulnerability class (**V1**) relies on the ability of the attacker to generate messages with malicious content (e.g.,

a fake location). By manipulating the information shared to other participants, the protocol logic can be exploited leading to safety or performance compromises. A related class of vulnerability (**V2**) concerns protocols where information that is critical to a sound decision is not considered, perhaps because it is not available, or is not exchanged. For example, the vehicles' lane position and platoon identification number are important parameters that we discovered were not considered when initiating a merge.

A third class of vulnerability (**V3**) relates to ambiguities that arise in *binding identifiers to vehicles*, the act of associating a detected physical information with a moving object such as a vehicle or pedestrian that is known through communication messages. Specifically, sensors can detect physical signals such as proximity to an object and mistakenly associate it with a different object in the message identifier space. For example, an attacker may pretend to be a platoon leader while a vehicle is attempting to join the platoon, a different vehicle may be mistakenly identified as the attacker/leader.

The next vulnerability class (**V4**) relates to under-specified or incomplete protocol logic. The application logic fails to consider corner cases such as the sudden loss of a platoon leader. In the reference CACC implementation [15], follower cars drive aimlessly if the platoon leader does not communicate with them. Ensuring the robustness of the protocol algorithm is essential for secure application.

The final vulnerability class (**V5**) arises when one object in the system delegates decisions to a malicious or compromised object, thus safety can be compromised. For example, trust is delegated to the platoon leader in CACC which enables arbitrary dangerous maneuvers that can cause crashes and blocking emergency vehicles.

5 Application level attacks on CACC

In this section, we present application layer attacks that attempt to exploit the functionality of the PMP implementation of CACC. These attacks were identified from a detailed code review of the PMP implementation. In each attack, we start with explaining the maneuver functionality and consider an attacker that participates in the protocol, sending messages in a way that passes the certificate based authentication and the application logic but results in disrupting the operation of one or more vehicles. We demonstrate the impact of these attacks in later sections.

5.1 Attack 1: Merge over large distances

If two platoons are traveling in the same lane and they are close enough while exchanging messages with each other, the PMP application allows them to merge to form one platoon for added efficiency. The application checks prerequisite conditions for the merge, such as, ensuring that the resulting

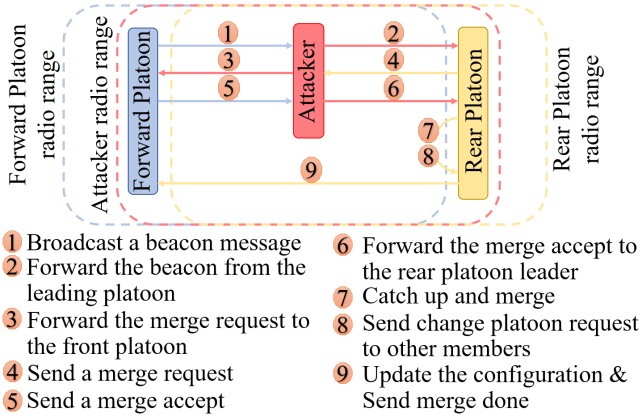


Figure 1: Attack scheme of distant merge attack

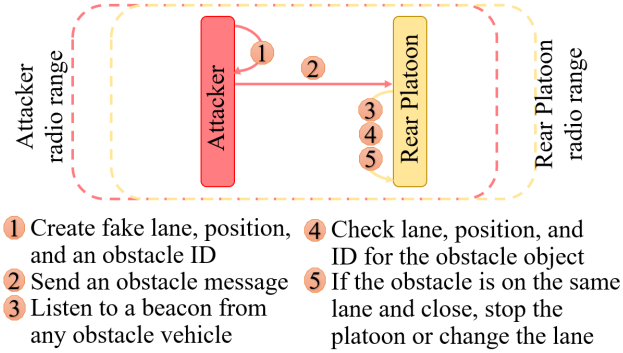


Figure 2: Attack scheme of the fake obstacle attack

combined platoon does not exceed the size limit. In our experiments, we found out that for two platoons to merge, the rear platoon must receive beacon messages from the front platoon. Then, it measures a certain distance to the last member of the front platoon using its ranging sensor. In our attack scenario, the attacker takes advantage of fake message contents (**V1**) and insufficient information (**V2**) vulnerabilities to target two platoons that are not within the communication range of each other. The attacker in this scenario is located between two platoons such that it can communicate with both platoons simultaneously and deceive ranging sensor by pretending that it is a member of the front platoon. For a farther distance, the attacker can have a sophisticated radio that can send and receive messages for a longer range.

The attack (Fig. 1) begins when the attacker replays the front platoon beacon messages to the rear platoon; since they are merely instantaneous replaying messages, the credentials on these messages are considered valid by the receiving vehicles. Upon receiving these beacons, the leader of the rear platoon will check to see if a platoon exists ahead by using its local sensors to look for a car from the front platoon in the lane ahead, which will be in this case, our malicious vehicle. The rear platoon will then speculate that the front platoon is

approaching and initiates merging if the new platoon size is under the predefined permissible threshold (i.e., size of the combined platoon is less than the maximum platoon size).

The rear platoon leader extracts the platoon ID of the front platoon from the beacon and sends a unicast merge request message to the front platoon (which is again relayed by the attacker). The front platoon leader, if it accepts the request, sends a unicast merge accept message, which the attacker then transmits back to the rear platoon. Upon receiving it, the rear platoon leader reduces its time-gap by increasing the speed of the whole platoon to the maximum limit to catch up. At this point, the attack impact shows up when the rear platoon increases its speed for a large distance degrading both safety and economy. Once the inter-platoon spacing becomes small, the rear platoon leader sends change platoon message to all its followers to change the platoon leader to the front platoon leader. Finally, the rear platoon leader sends a merge done message to front platoon leader and changes its state from leader to follower.

5.2 Attack 2: Fake Obstacle Attack

A platoon may have automatic incident detection enabled; with this option, the platoon can receive and rapidly react to an obstacle message. Upon encountering an obstacle or accident in its lane, a vehicle will come to a stop and send an obstacle message with its position to any oncoming vehicles, allowing them to stop or change their lanes when they arrive at the location of the incident. In this scenario, the malicious vehicle exploits the fake content (**V1**) vulnerability and creates a false obstacle message with a specific location in the lane, forcing incoming platoons to slow down until they stop or change lanes. The attack scheme is shown in Fig. 2. The fake obstacle attack affects the speed of the platoon and this rapid deceleration can affect safety. The presence of an obstacle is impossible to validate by a distant platoon. Note, that it is possible to combine this attack with *Attack 1* to attempt to create an accident by first speeding up the cars and then forcing them to stop quickly.

5.3 Attack 3: Merge across different lanes

In this scenario, we attack two platoons, within the communication range of each other, that are traveling in separate lanes. Critical variables such as lane number and other surroundings information for each vehicle are neither communicated nor checked (**V2** and **V4** vulnerabilities). The attacker can look for a slow platoon in front and try to merge it with a faster platoon from a different lane to slow down traffic flow.

The attack (Fig. 3) starts when the malicious vehicle is in front of the rear platoon, and sends messages pretending to be a part of the other platoon (in another lane). This can be done by manipulating the platoon ID parameter in Basic safety message. The rear platoon will see the attacker vehicle

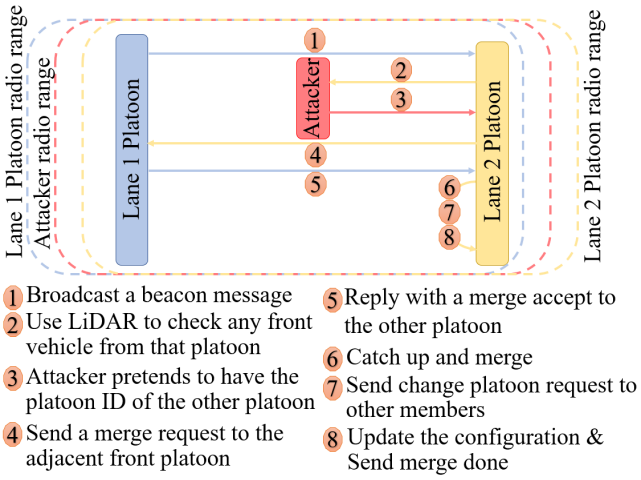


Figure 3: Attack scheme of merging across lanes attack

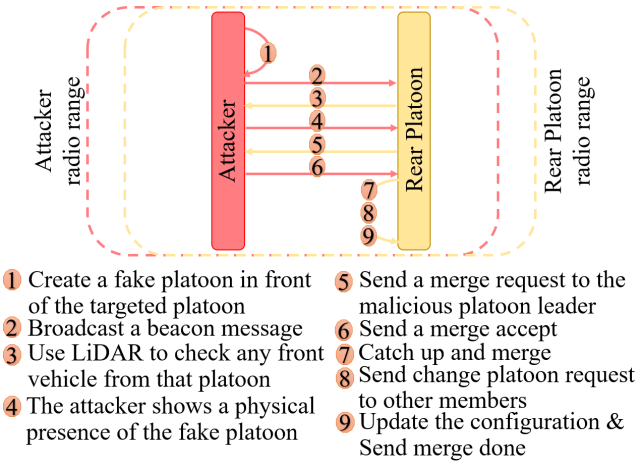


Figure 4: Attack scheme of platoon takeover attack

using its LiDAR sensor and assumes that the attacker is part of the platoon (**V3**). Information such as Lane ID is neither communicate nor checked. It then begins a merging maneuver. As consequence, the adjacent leading platoon leader sends a merge accept message. As a result, the rear platoon leader increases the speed of the platoon to catch up. Afterwards, the attacker leaves its location and the rear platoon leader sends change platoon to all its followers.

5.4 Attack 4: Platoon Takeover

This attack is conceptually similar to the *Attack 3* except that there is only one platoon (the rear platoon), with the attacker attempting to become its leader. The attacker counts on different vulnerabilities but mainly on the fake message contents (**V1**) vulnerability by pretending to be the leader of the fictitious front platoon by generating any logically consistent description of the front platoon such as the locations and

speeds of a fake platoons' members in front of the victim platoon. The attacker transmits the fake messages for each false vehicle of the fake platoon. The rear platoon leader will notice the attacker through the LiDAR sensor and initiate a merging maneuver since it believes that this is the platoon in front of it that it listens to. The attacker responds to all requests from the rear platoon. This leads to the completion of the merging process. The platoon is now under the attackers' control and can be manipulated in a dangerous manner as we show in Section 6.1, exploiting the trust delegation (**V5**) vulnerability. We show the steps of this attack are shown in Fig. 4.

6 Experimental Attack Scenarios and Results

In this section, we first describe the simulation set up used in the experiments. We then present an experimental evaluation of the proposed attacks and evaluate their impact on the traffic system with respect to safety and performance. Given the limited availability of deployed CV applications, and the closed nature of these systems, we elected to evaluate the attacks using simulation. We used VENTOS (VEhicular NeTwork Open Simulator), an extension of Veins [27]. Veins integrates a C++ simulator for studying vehicular traffic flows, collaborative driving, and interactions between vehicles and infrastructure with another simulator which models communication through a DSRC-enabled wireless communication. Veins combines two widely used simulators, Simulation of cars/physics simulator (SUMO) [5] and OMNET++ [3]. SUMO is an open-source road traffic simulator developed by the Institute of Transportation Systems at the German Aerospace Center and serves as the traffic flows physics simulator. This framework has been used in hundreds of studies from academia, industry, and the government (a partial list can be found on the project [8]). VEINS uses SUMO's Traffic Control Interface, TraCI, to communicate simulation commands to it. OMNET++ is an open-source simulation package and carries out the wireless communication simulation. We configure it to use the models for the IEEE 802.11p [21] protocol, a standard adopted for V2V communication. We use Wave Short Message Protocol (WSMP) to carry beacon and micro-command messages. These messages are directly sent to the data-link layer which uses continuous channel access based on IEEE 1609.4.

6.1 Dangerous Attack Demonstrations

First, we demonstrate the potential impacts of the attacks using two specific scenarios, one causing a collision and the second interfering with and delaying an emergency vehicle.

Causing a Collision: In this attack, the followers of a platoon that is controlled by a compromised leader, fail to see and stop for stationary or slower vehicles. The malicious car may have acquired leadership of the platoon using the platoon takeover attack. The attacker can suddenly veer out of a lane without informing the followers to slow down or change lanes. The followers' braking systems may not be able to stop if an obstacle appears immediately in their path. We can see the sudden stop then collision at the time 60s for the victim vehicles in Fig. 5. After investigating this scenario in detail, we discovered that vehicles in the platoon were not keeping a safe distance between each other. Instead, they were delegating trust (**V5**) to the platoon leader (the attacker), trusting that the leader will maintain safe separation from any obstacles.

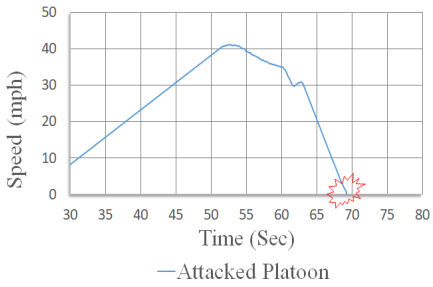


Figure 5: Speed profile in collision attack

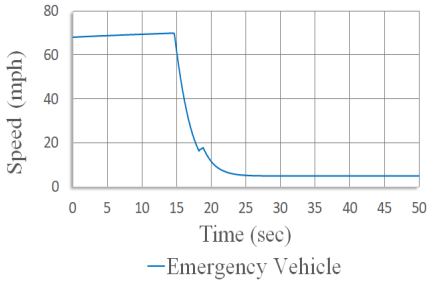
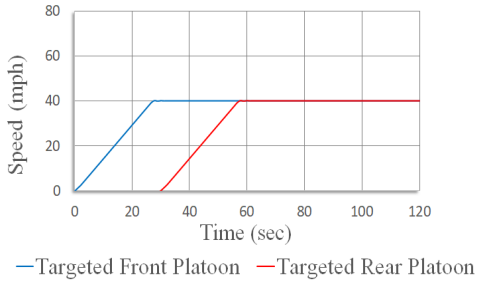


Figure 6: Speed profile in Emergency Vehicle Attack. The attacker slows down the emergency vehicle from 70 to 5 mph

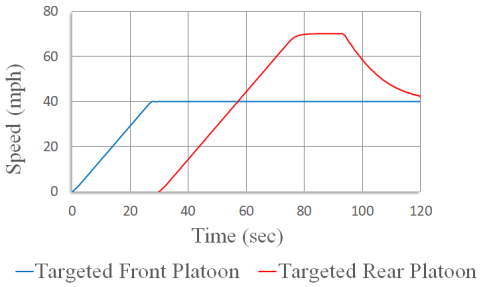
Emergency Vehicle Interference: We again start with the attacker using the *Platoon Takeover* attack, described in Section 5.4. The attack is comprised of Attack vector V5 (untrusted deligation). The attacker slows down the whole platoon then makes some followers move to another lane. If an emergency vehicle (police or ambulance) is coming fast in that lane, a slow vehicle on the same lane will make it much slower or even stop it, as shown in Fig. 6. This can cause catastrophic slowdowns in real life (e.g., potential loss of life). Other approaches to delay an emergency vehicle can be devised, for example, using the merge across different lanes attack.

6.2 Isolated Attack Scenarios

In this set of experiments, we investigate vehicle performance after implementing the four different attacks described in Section 5 isolating the impact on just one or two targeted

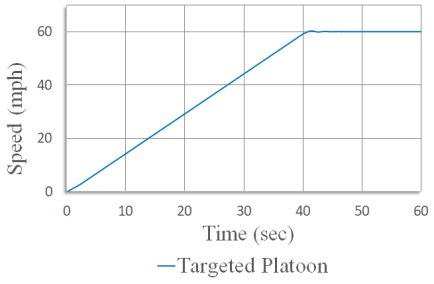


(a) Before attack

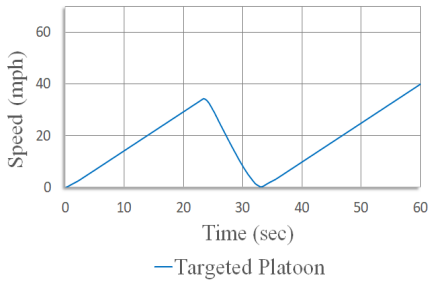


(b) After attack

Figure 7: Speed profile in Attack 1 (distant merge)



(a) Before attack

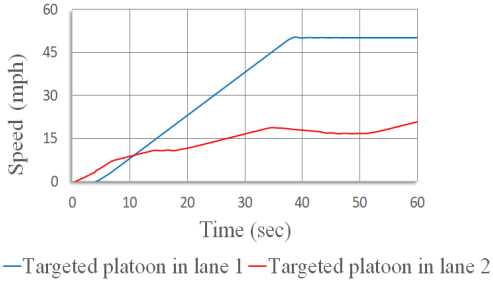


(b) After attack

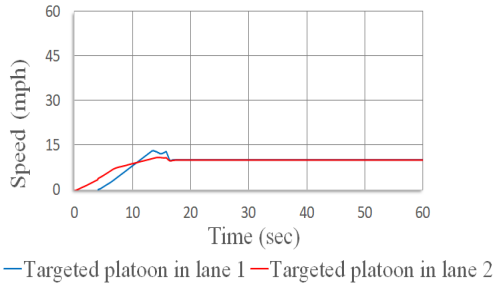
Figure 8: Speed profile in Attack 2 (fake obstacle attack)

platoons. These scenarios allow us to evaluate the isolated impact of the attacks.

Attack 1– Distant Merging attack: Our intention in this attack is to make some platoons go to the catch-up process where they speed up abnormally for some time potentially degrading both safety and efficiency. Fig. 7a shows the average speed of two platoons in the scenario in the absence of an attack. The rear platoon starts a little later, but both platoons accelerate to 40mph before cruising at that speed. Fig. 7b shows the behavior of the platoons in the presence of the attack. In this case, the rear platoon accelerates aggressively, reaching the maximum velocity, in an effort to catch up with the front platoon.



(a) Before attack



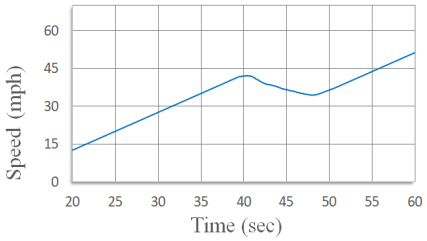
(b) After attack

Figure 9: Speed profile in attack 3 (merging across lanes)

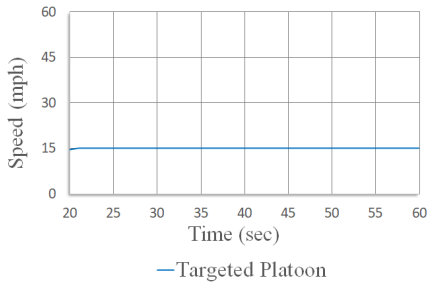
Attack 2– Fake Obstacle attack: From Fig. 8a, we see a platoon of 3 cars accelerating to 60mph. After initiating the attack starting around time 20s, we can notice how the platoon suddenly comes to a halt as shown in Fig. 8b. This occurs for a certain time then the platoon changes the lane and accelerates again, but the attack can be repeated.

Attack 3– Merging platoons across lanes: In this scenario, two platoons travel on different lanes where the front platoon is slower than the rear one. The attacker realizes that both platoons are close to each other and locates itself in front of the rear platoon. Next, the attacker initiates the merge maneuver as described in *Attack 3*. When the attack succeeds, all the members of the rear platoon will follow the front platoon (despite being in a different lane) and travel according to its speed as shown in Fig. 9. In this case, the lower speed platoon slows down the traffic flow. In another case, the rear platoon may be tricked to go faster than the optimal speed for the lane, compromising safety.

Attack 4– Platoon Takeover Attack: The attacker starts with sending different beacon messages pretending that they come from a front platoon. Once the platoon finds that the leading vehicle on the same lane is the last platoon member that it listens to (through its LiDAR sensor), it will then start



(a) Before attack



(b) After attack

Figure 10: Speed profile in attack 4 (platoon takeover attack)

the merging process. After the merging succeeds, the attacker now acts as a platoon leader and controls this platoon in any way it desires within the platoon operational parameters. For this example attack, the attacker decreases the platoon velocity and then repeatedly changes the lane of the platoon in order to affect as many lanes as many as possible. Fig. 10 shows the platoon speed changes.

6.3 Attacks within traffic scenarios

Next, we evaluate the impact of the attacks when applied as part of an active traffic scenario. We use different metrics to quantitatively analyze the effects of the attacks on *Mobility* and *Safety*. For mobility, we use two metrics: (1) **Average speed** of vehicles is a common metric for mobility; and (2) **Flow** of traffic, is defined as the number of vehicles passing a point on the road in a given time. To measure safety, we also use two metrics: (1) **Average speed difference** between consecutive vehicles measures the differences in speed among vehicles. This metric is known to correlate with the onset of collisions and near-collisions; and (2) **Time-to-Collision (TTC)** [23] is metric for safety which measures the time taken for a vehicle to collide with the vehicle in front of it, should they maintain the same speed. TTC of vehicle i at instant t can be calculated as follows,

$$TTC_i(t) = \frac{V_i(t) - V_{i-1}(t) - l_i}{V_i(t) - V_{i-1}(t)}$$

here, $V_i(t)$ stands for the speed of the vehicle i at instant t and l_i is the length of the vehicle i.

California Department of Transport provides real time traffic condition through Performance Measurement System [4] by using various sensors installed in the state's most highways sections. We use data from a section of the highway I-5 in south California and generate scenarios with vehicles entering stochastically following the observed distribution. Each scenario, ran for 5 minutes, simulates the entrance of traffic into a highway section of length 6 miles. We assume that all vehicles are CV enabled to avoid making assumptions on the interactions of CV and non-CV vehicles. We configure about 25% of the vehicles to form platoons of different sizes. The maximum speed for the road is 70 mph. The communication range for each vehicle is 300 meters. Each road has five lanes and approximately evenly spaced road side units (RSU) such that all points in the highway are in range with at least one RSU.

Attack 1–Distant Merging Attack: Fig. 11a shows the effects of attack 1 on the average speed, flow, average speed difference, and average TTC for the scenario. The attack causes an increase in average speed and flow of traffic. Even though the flow of vehicle increases by a small amount, the attack causes vehicles under attack to travel at a much higher speed, thus compromising safety, which is reflected by the increased average speed difference and reduced TTC. Even though the flow of vehicle increases by a small amount, distant merge attack causes vehicles under attack to travel at a much higher speed, thus compromising safety.

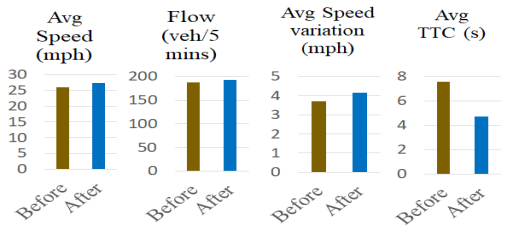
Attack 2– Fake Obstacle Attack: Fake obstacle attack causes the traffic to slow down potentially abruptly, similar to the slow down due to road site construction. Thus, it has slight adverse effect on safety, with increased average speed difference and TTC, but a large effect on the mobility, with decreased average speed and flow, as depicted in Fig. 11b.

Attack 3– Merging across lanes: In this attack, the attacker connects the flow of traffic of two or more lanes, forcing a faster platoon to slow down. The effect of the attack is shown in Fig. 11c. Average speed difference increases only slightly, while TTC increases, leading to a marginal impact on safety. However, the flow of the traffic is severely hindered which is shown by the steep drop in average speed and flow.

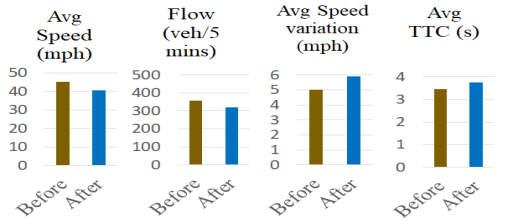
Attack 4– Platoon takeover: In this attack, the attacker takes over the control of a platoon and can control it fully. This is the most dangerous form of attack that the attacker can carry out. Although different arbitrary maneuvers are possible once the attacker controls the platoon, we went with a speed reduction and repeated lane change maneuvers. Both safety and mobility metrics are highly affected by this attack, as seen in Fig. 11d.

7 Potential Mitigation

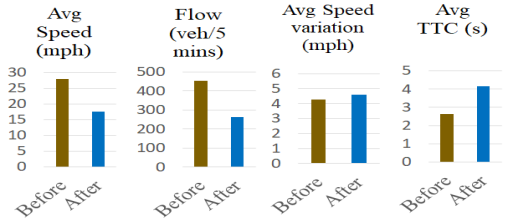
Our eventual goal is to develop a defense approach that is automated and can mitigate the vulnerability classes we identified in Table 1, thus making the protocol logic more secure in a principled way. The general defense approach relies on



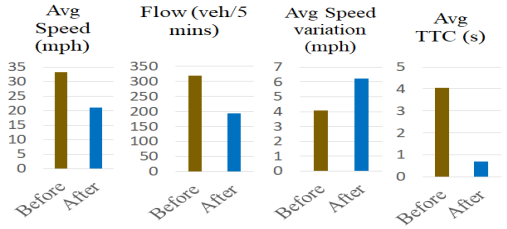
(a) Impact of attack 1: Distant merge



(b) Impact of attack 2: Fake Obstacle



(c) Impact of attack 3: Merge across lanes



(d) Impact of attack 4: Platoon takeover

Figure 11: Impact of attacks

augmenting the information available to vehicles with a redundant source of information that enables detection of incorrect or malicious information, and makes the protocol logic more robust. If such a source of redundant information is available, the veracity of the exchanged messages can be checked before conducting critical actions within a maneuver, thus addressing **V1** and **V2** vulnerabilities. To give an example, if a merge is attempted with a far-away platoon, the requested platoon should check if the distance of the front platoon is within the merge range; previously, this was assumed from the fact that the messages were received from the front platoon, an assumption that can be exploited by an attacker that replays a message (effectively extending its reach) or to use higher power radio to increase its range. Moreover, this defense substantially reduces the opportunities for **V3** attacks since it becomes more difficult to create wrong bindings between message sources and other physical objects. This check would defeat the replay attack that allows the adversary to initiate a merge. **V4** can be addressed by in depth protocol testing and analysis. Finally, **V5** can be addressed by either avoiding trust delegation or verifying delegated decisions.

We collect complementary information through a reliable sensory system to protect against fake message contents (**V1**). Validating protocol components by linking message contents and redundant sensor data is also desirable for a reliable decision. The consistency of the application and environment constraints using a robust algorithm need to be considered to prevent a message with clearly unfeasible information to be acted on and ensure that the resulting action is consistent with the protocol logic. If everything checks out, a final decision will be assigned to protocol controller to lead the required action.

7.1 Preliminaries and Assumptions

RSU: Defense components infrastructure: The main component that we rely on in our scheme is the road side unit (RSU), where its hardware and software components are specified by US DOT [26]. The RSU is a more sophisticated and more protected component of the system deployed and managed by the infrastructure provider, making it an attractive component to root defenses. It is expected to operate unattended in harsh outdoor environments for extended periods of time (typical Mean Time Between Failures of 100,000 hours). It detects and auto-recovers from minor software failures, transient power spikes, and power interruptions. We consider a case where RSUs are reachable from any point on the highway as a proof of concept, but the protocol can be made to act conservatively in Safe mode when RSU are not reachable.

We note that without relying on the RSU, the alternative is to reach consensus between the different cars which is an interesting possibility. A naive implementation could be too costly to achieve on-demand, and therefore we elected to root our defenses in the RSU.

Safe mode and functionality of RSU: We identify a safe operation mode for platoons with respect to any maneuver or protocol state. The goal of the safe mode is to be used as a cautious behavior when protocol exchanges are in progress, or when a decision cannot be made. For example, the platoons could either maintain their speed or slow down and wait for confirmation after sending a maneuver request. The defense proceeds by having the RSU check the the proposed action against the configuration of the platoon (e.g., the location of each member of the relevant platoons from all basic safety messages (BSMs) it collects). The RSU uses, as a source of redundant information, a video tracking system to track the vehicle locations. The system also maps any incoming messages to vehicles based on the geographic information to check the consistency of messages being sent by any particular vehicle.

Algorithm 1 Pre-Approval protocol for Platoon Leader

```
1: procedure PRE-APPROVALPROTOCOL
2:     SendManeuverRequestToRSU()
3:     Change to SAFE mode        ▷ Wait for RSU response
4:     loop:
5:     if Disapproval Received then return AbortManeuver()
6:     if Approval Received then return StartManeuver()
7:     if Time Out Exceeded then return Exit-loop
8:     goto loop                  ▷ Time Out NOT Exceeded
9:     StartBackupProcedure()
```

Other sources of redundancy are also possible, for example, exchange of past information from nearby RSUs for vehicle tracking, or alternative real time sensors. Our proposed video tracking system is feasible: many vehicles tracking systems using video cameras have been proposed [29], [28]. We would next see how the defense would work for the previous attacks.

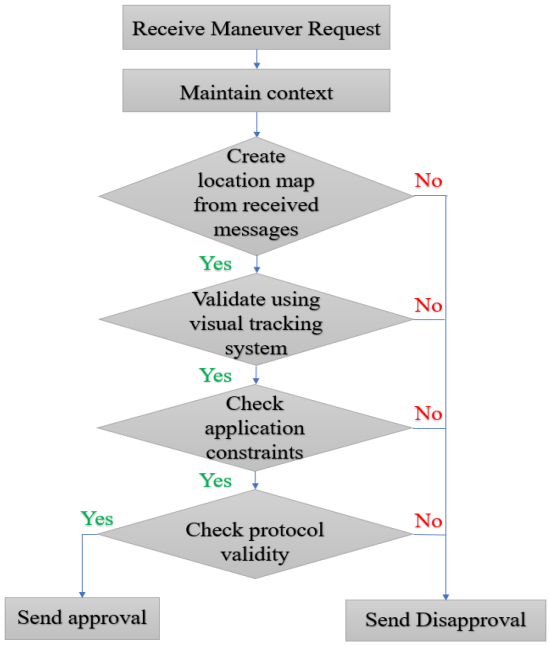


Figure 12: Pre-Maneuver Protocol process for RSU

7.2 Defense overview

Defense against Merging attacks: For *Attacks 1, 3*, and *4*, the defense starts by allowing the back platoon to send a merging request to RSU. After receiving the maneuver request, the RSU verifies the relevant information. Then, it tests if the merge process is applicable or not by inspecting the constraints between the platoons such as making sure that the distance between them is within the permissible range. If all checks pass, an approval reply is sent to the two platoons to start merging. If the maneuver confirmation is received and leader, for any reason does not exist, the platoon members can start a voting process where they study the collected BSMs and check its neighbor vehicles through its sensors to choose their leader to control the maneuver.

Defense against Obstacle attacks: For *Attacks 2*, RSU carries out the same steps regarding requesting a maneuver. For this scenario, it checks specifically if the obstacle and the incoming platoon are in the same lane or not and, if yes, the distance between them. Then, the RSU will send an approval reply to stop the coming platoon or change its lane. In the meantime, the traveling platoon leader will go to the safe mode where it moves within the safety speed limit which we defined here to be below 20 mph. Generally, it is sufficient to ensure the ability to stop in case the obstacle message is confirmed. If the platoon does not receive any confirmation for the obstacle maneuver until the obstacle location, it can start the backup protocol where it can stop or change lane. Fig. 12 shows the general protocol for the RSU. Algorithm 1 shows the steps for the platoon leader.

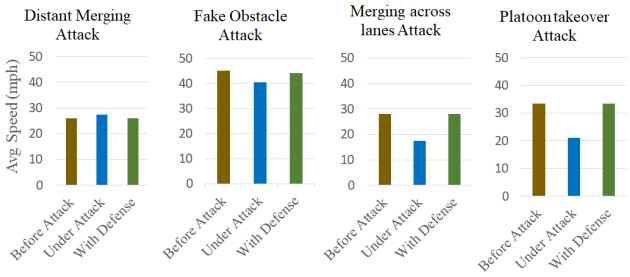


Figure 13: Effect of defense on the studied attacks

7.3 Evaluation

We implemented the defense logic within the simulator. We emulate the video tracking by using the ground truth value of the location and adding Gaussian noise to it with a mean of 2 meters. We augmented the application with the defense by following the mitigation steps discussed above. Fig. 13 demonstrates that with the defense in place, the attack impact is mitigated from all attacks other than the fake obstacle attack where it has a minor effect. The effect is due to the delay in confirmation from the RSU, during which the safe mode reduces performance whether there is a real obstacle or not. This mitigation's approach will also be able to stop the dangerous attacks described in Section 6.1. This is due to the fact that those attacks are bases on the basic attacks demonstrated in Section 5 but used in specific scenarios.

7.4 Discussion

A concern with any defense strategy that requires additional operations is delays in making decisions, while information is validated. However, we believe that the redundant information can be prepared proactively so that the check is often local. Moreover, it is critical to deploy the safe backup operation while decisions are being taken in any cyber-physical system, prioritizing safety over performance.

The approach heavily relies on the visual tracking system and sensors for more reliable decision, which may not be available in all vehicles and in some areas. Thus, we accept that it is a strong assumption on our part to assume that such redundant data will always be available to the decision making system. We can see from Fig. 13 that safe mode does not significantly degrade performance in CACC application. Nevertheless, we will carry out analysis and performance measurements on other CV applications to justify this statement in the future. In the future work, robust algorithms may be employed to detect all the different attacks in the early stages.

8 Vulnerabilities in other protocols

In this section, we analyze other protocols and classify the vulnerabilities using the attack vectors defined in Table 1. We performed code reviews of two protocols available on the US DOT open source CV protocol repository [13]: (1) Intelligent Intersection Management and (2) Eco-Traffic Signal Timing.

Intelligent Intersection Management has shown great potential in improving transportation efficacy especially for autonomous vehicles where it connects with them wirelessly and schedules their intersection crossing steps. In [17], they proposed to use existing infrastructure-side sensors to stop malicious messages. For example, vehicle detectors buried underneath the stop bar of each lane can be used to measure aggregated traffic information. After analyzing the scheme, we found out that malicious messages can still be sent to manipulate the application and increase total delay time. This is due to inadequate identifier binding (**V3**) vulnerability, where sensors do not correlate the messages with the vehicles and do not give the exact location for each vehicle.

Eco-Traffic Signal Timing application aims to improve traffic signals delays thus reducing environmental impact. It processes real-time and historical CV data at signalized intersections to reduce fuel consumption and overall emissions. In this application, we discovered that vehicle trajectory data can be subjected to fake message contents (**V1**) and inadequate identifier binding (**V3**) vulnerabilities. We were able to implement exploits to manipulate the timing phase for any lane based on sending malicious vehicles information.For both applications, the defense principles we introduced can be adapted to mitigate these vulnerabilities.

9 Related work

Chen et al. [17] performed a security analysis on a system called Intelligent Traffic Signal System (I-SIG). In this system, a real-time vehicle trajectory data is sent through the dedicated short-range communications (DSRC) technology that is acquired by any CV. This data is then used to control the sequence and duration of traffic signals. The system that was attacked includes real deployments at road intersections in some cities such as Anthem, AZ, and Palo Alto, CA. In these deployments, it enhanced the traffic by reducing total vehicle delay by 26.6%. The paper presented an attack that can cause the traffic mobility to be 23.4% worse than that without adopting I-SIG. The attack consists of a packet spoofed to tell the I-SIG of a vehicle approaching from a long distance, causing the traffic light to wait for it, while holding up traffic from other directions. The authors suggested a possible defense that considers scheduling over multiple periods.

Amoozadeh et al. [14] performed a security and vulnerabilities risks analysis related to the VANET communication in CV in specific applications including cooperative adaptive cruise control application (CACC). They focused mainly on how to attack a single platoon. They considered a CACC vehicle stream that is moving in a straight single-lane highway where all the vehicles use a simple one vehicle look-ahead communication scheme. They did not consider the security credential management system (SCMS) in their simulation. In their work, they examined existing countermeasures, and explored the limitations of these methods and possible ways to lighten negative effects.

Dominic et al. [18] presented a risk assessment framework for autonomous and cooperative automated driving was conducted to define a reference scheme for automated vehicles, and to describe the new attack surfaces and data flow. They used recent automotive threat models and introduced a novel application based threat enumeration and analysis approach that is able to address different automated vehicles applications across all levels of automation. They also established a framework with an example application assessment. In their work, they concluded that their results would guide the design of the secured automated driving architectures that will certainly and quickly become necessary.

The Intelligent Transportation Systems Joint Program Office (ITS JPO) [7] worked with its partners with a fund nearly $25 million to support a foundational vehicle cyber security threat assessment for CV applications. Their work includes designing, developing, and operating the security credential management system (SCMS) for the CV Safety Pilot evaluations were conducted in some cities such as Ann Arbor, Michigan, as well as the current US DOT CV pilot studies in NYC [11], Tampa [12], and Wyoming [10]. They developed certification practices to check equipment prior to implementation in the Safety Pilot to ensure that they meet cyber security requirements. The program is also working to ease provid-

ing the certification services for different industries. Finally, one of the goals is to improve the best practices for handling foundational electronics control and reliability cyber threat information for existing vehicle fleets.

10 Concluding Remarks

Connected Vehicles (CVs) is an emerging field in transportation that is garnering interest from the US DOT because it promises to bring about a new age of transportation where vehicles and transportation infrastructure are all interconnected wirelessly. However, many applications are still not considering their security vulnerabilities. This paper shows that one of the most complete reference implementations of a CVs protocol (for Cooperative Automatic Cruise Control) is vulnerable to attacks of many types, even under a threat model that considers the state-of-the-art SCMS certificate based security standard being developed for these applications. These attacks that exploit the vulnerabilities of these communication protocols, may lead to a complete reversal of the benefits made by CVs, and as such, they have a ways to go before it is reliably safe from attacks. We demonstrated these attacks in simulation and showed their impact on safety, performance, and economy of the traffic.

It also introduces a defense scheme that places vehicles in a safe mode while they check the consistency of received information against an estimate of the local traffic state constructed through video analytics at the road side unit. Finally, we showed that the proposed defenses are able to mitigate all the attacks we introduced, making it a promising approach to support security in CV applications.

Acknowledgements

This material is partially supported by the National Science Foundation (NSF) grant CNS-1646641 (CNS-1839511) and CNS-1724341. It is also partially supported by UC Lab Fees grant LFR-18-548554. All opinions and statements reported here represent those of the authors.

References

[1] Coordination of Automated Road Transport Deployment for Europe. Accessed Dec 2016 from https://connectedautomateddriving.eu/wp-content/uploads/2017/02/20161216_CARTRE_MS_Workshop_v1.1-1.pdf.

[2] CV Pilot Deployment Program. Accessed from https://www.its.dot.gov/pilots/cv_pilot_apps.htm.

[3] OMNeT++. Accessed from https://www.omnetpp.org/.

[4] PeMS - Caltrans Performance Measurement System. Accessed Aug 2018 from http://pems.dot.ca.gov/.

[5] SUMO - Simulation of Urban Mobility. Accessed from http://sumo.dlr.de/index.html.

[6] The Tesla Semi-trailer truck as It relates to platooning. Accessed Dec 2017 from http://adamkwitko.com/the-tesla-semi-as-it-relates-to-platooning/.

[7] USDOT: Security Credential Management System (SCMS). Accessed from https://www.its.dot.gov/factsheets/pdf/CV_SCMS.pdf.

[8] VEINS - Vehicles in Network Simulation. Accessed from http://veins.car2x.org/.

[9] VENTOS - VEhicular NeTwork Open Simulator. Accessed from http://maniam.github.io/VENTOS/.

[10] Connected Vehicle Pilot Deployment Program, Wyoming, 2018. Accessed August 2018 from https://www.its.dot.gov/pilots/pilots_wydot.htm.

[11] Connected Vehicle Pilot Project, New York City, 2018. Accessed August 2018 from http://www.cvp.nyc.

[12] Connected Vehicle Pilot Project, Tempa, 2018. Accessed August 2018 from https://www.tampacvpilot.com/.

[13] The Open Source Application Development Portal for Federal Highway Administration, USDOT, 2018. Accessed Aug 2018 from https://www.itsforge.net/index.php/community/explore-applications/for-search-results#/30/63.

[14] 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. *IEEE Communications Magazine*, 53(6):126–132, June 2015.

[15] Mani Amoozadeh, Hui Deng, Chen-Nee Chuah, H Michael Zhang, and Dipak Ghosal. Platoon management with cooperative adaptive cruise control enabled by vanet. *Vehicular communications*, 2(2):110–123, 2015.

[16] Matthew J Barth, Guoyuan Wu, and Kanok Boriboonsomsin. Intelligent transportation systems and greenhouse gas reductions. *Current Sustainable/Renewable Energy Reports*, 2(3):90–97, 2015.

[17] Qi Alfred Chen, Yucheng Yin, Yiheng Feng, Z Morley Mao, and Henry X Liu. Exposing congestion attack on emerging connected vehicle based traffic signal control. In *Proceedings of the 25th Network and Distributed System Security Symposium (NDSS'18), San Diego*, 2018.

[18] R. Eustice D. Ma Di D. Dominic, S. Chhawri and A. Weimerskirch. Risk assessment for cooperative automated driving. *Proceedings of the 2Nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, pages 47–58, 2016.

[19] Fluidmesh Networks LLC. DSRC Roadside Unit. Accessed March 2019 from https://www.fluidmesh.com/dsrc-roadside-unit/.

[20] John Harding, Gregory Powell, Rebecca Yoon, Joshua Fikentscher, Charlene Doyle, Dana Sade, Mike Lukuc, Jim Simons, Jing Wang, et al. Vehicle-to-vehicle communications: readiness of V2V technology for application. Technical report, United States. National Highway Traffic Safety Administration, 2014.

[21] Hannes Hartenstein and LP Laberteaux. A tutorial survey on vehicular ad hoc networks. *IEEE Communications magazine*, 46(6), 2008.

[22] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. Mobile edge computing—a key technology towards 5g. *ETSI white paper*, 11(11):1–16, 2015.

[23] Michiel M Minderhoud and Piet HL Bovy. Extended time-to-collision measures for road traffic safety assessment. *Accident Analysis & Prevention*, 33(1):89–97, 2001.

[24] University of Arizona, SCSC Econolite University of California PATH Program, Savari Networks Inc, and Volvo Technology. Multi-Modal Intelligent Traffic Signal Systems (MMITSS), Concept of Operations, 2012. Accessed May 2018 from http://www.cts.virginia.edu/wp-content/uploads/2014/05/Task2.3._CONOPS_6_Final_Revised.pdf.

[25] U.S. Department of Transportation. The Connected Vehicle Reference Implementation Architecture. Accessed March 2019 from https://local.iteris.com/cvria/html/applications/applications.html.

[26] Frank Perry, Kelli Raboy, Ed Leslie, Zhitong Huang, Drew Van Duren, et al. Dedicated Short-Range Communications RoadSide Unit Specifications. Technical report, United States. Department of Transportation, April 2017.

[27] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, January 2011.

[28] Yao-Jan Wu, Feng-Li Lian, and Tang-Hsien Chang. Traffic monitoring and vehicle tracking using roadside cameras. *2006 IEEE International Conference on Systems, Man and Cybernetics*, 6:4631–4636, 2006.

[29] Jingxin Xia, Wenming Rao, Wei Huang, and Zhenbo Lu. Automatic Multi-Vehicle Tracking using Video Cameras: An improved CAMShift approach. *KSCE Journal of Civil Engineering*, 17:1462–1470, 09 2013.