Attack Modeling Methodology and Taxonomy for Intelligent Transportation Systems

Anthony Bahadir Lopez, *Member, IEEE*, Wen-Long Jin, *Member, IEEE*, and Mohammad Adbullah Al Faruque, *Senior Member, IEEE*

***Abstract*— With newer technologies, the embedded hardware and software in traditional vehicles and traffic control infrastructure continue to become more interconnected and more vulnerable. To assist in dealing with existing and potential vulnerabilities, we present a novel attack modeling methodology, taxonomy, and metrics (relative average waiting time, average network flow, impacts and rate of changes) to model, simulate, and meaningfully evaluate the security of Intelligent Transportation Systems. We implement our work in two different architectures: 1) Newell's Car-Following Model with Bounded Acceleration (the BA-Newell Model) in Matlab and 2) Intelligent Driver Model in Veins. Our code is entirely open-sourced and will be maintained so that the ITS community may use it as a tool. We observe that the architectural-related metric values for sample attack simulation results are similar and transferable; where, for example, the rate of change values have range of average distances 1.8-3.5% for network flow impact and 3.3-9.6% for wait time impact.**

***Index Terms*— Intelligent Transportation Systems (ITS), intelligent vehicles, communications technology, computer networks, network security.**

I. Introduction and Related Work

MASSIVE deployment of embedded systems including various sensors, on-board and road-side computing units, wireless communication among vehicles and infrastructure, and intelligent algorithms are changing the transportation sector [1]–[5]. Due to these technologies, engineers are able to provide autonomy and connectivity in transportation control systems. Therefore, this new paradigm, known as Intelligent Transportation Systems (ITS), is bringing new opportunities to solve transportation system challenges regarding traffic congestion, energy waste, vehicle emissions, and traffic accidents [6]. Today, advanced sensors and wireless vehicular communication (V2X) enable advanced algorithms for traffic management such as autonomous control, Connected Adaptive Cruise Control (CACC), collision detection and avoidance, Advisory Speed Limit (ASL), route guidance, and more [7].

Manuscript received 25 June 2020; revised 8 June 2021; accepted 1 October 2021. Date of publication 3 November 2021; date of current version 9 August 2022. The work of Anthony Bahadir Lopez was supported by the National Science Foundation Graduate Research Fellowship Program under Grant DGE-1839285. The Associate Editor for this article was E. Kaisar. *(Corresponding author: Anthony Bahadir Lopez.)*

Anthony Bahadir Lopez and Mohammad Adbullah Al Faruque are with the Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA 92697 USA (e-mail: anthl10@uci.edu; alfaruqu@uci.edu).

Wen-Long Jin is with the Department of Civil and Environmental Engineering, Institute of Transportation Studies, University of California at Irvine, Irvine, CA 92697 USA (e-mail: wjin@uci.edu).

Digital Object Identifier 10.1109/TITS.2021.3123193

As discussed in works [8]–[12], with these newer technologies come unforeseen safety and security concerns. Some of these concerns were recently revealed when traffic controllers used in almost all of the states in the US were found to be remotely hackable and controllable by an attacker [8], [13]. In addition to traffic control systems, connected autonomous vehicles provide many security and safety concerns, where effects of attacks on the peripherals or the Electronic Controller Units (ECUs) may cause congestion, but more importantly may endanger passengers and passersby [14]–[17].

II. Overview and Contributions

With this work, we are the first to develop a general attack modeling methodology and taxonomy regarding potentially targeted components in an ITS. We provide unique attack impact metrics and evaluate them with different car-following models and simulation tools. The taxonomy and attack models are implemented on the use case of V2X Advisory Speed Limit (ASL) control - which has never before been studied in terms of security, nor thoroughly implemented/studied in general. Implementations of an ITS application simulation may vary substantially and therefore establishing a foundation to do so is timely and critical in this rapidly developing technological age.

Historically, ASL has been performed using only physical signs on the road to help reduce speeds. With the advent and adoption of V2X and 5G, traffic management applications like ASL will be strongly improved. In the paradigm of ITS, ASL is one of the key upcoming applications whose impacts have not yet been analytically or experimentally studied in terms of security [5]. V2X ASL focuses on taking information from induction loop sensors and the connected vehicles arriving to an intersection from upstream to then advise the connected vehicles the maximum velocity (v_{asl}) that they should follow to ideally arrive at the earliest green or yellow time. This substantially reduces the average waiting time aka stopping time, improves the overall average network flow-rate, and subsequently lowers environmental costs from greenhouse gas emissions and even improves driver attitudes [6]

The following items will be the order of contents of this paper and also the summary of our contributions:

- Design and implementation of an ITS use case known as V2X Advisory Speed Limit control.
- Usage of two different simulation architectures composed of the Ring Road Network Model with

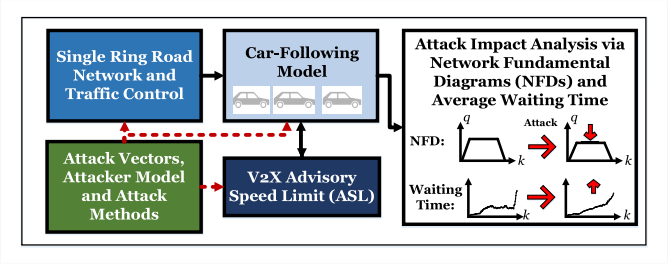

Fig. 1. Methodology overview. The dashed red arrows correspond to the points of possible attack injections and the solid black arrows are the dependencies between the models. The Car-Following Model and ASL blocks may be switched with equivalent components to evaluate other models and ITS applications. Here, q is the average network flow in *veh/s* and k is the average vehicle density in *veh/m*).

1) Newell's Car-Following Model with Bounded Acceleration (BA-Newell Model) and Matlab and 2) Intelligent Driver Model and Veins.

- Meaningful performance and attack impact metrics that may be transferable in different use cases, car-following models, and simulation tools.
- Creation of attacker and attack modeling taxonomy to serve as a guide for ITS security analysis.
- Evaluation of attack impact metrics using various attacker profiles, attack types, and timing parameters.
- Development of open source code for both architectures to allow others in the ITS community to utilize or improve upon our work.[1]

III. ITS Attack Modeling Literature Review

With respect to attack modeling in traffic control systems and connected vehicles, past studies have studied attacks on unique use cases and network models [18]–[20]. In contrast, we use the Single Ring Road Network Model for its simplicity and ability to abstract more complex network and system models. The attack vectors proposed in these works are the same as those in the past studies, aka, the methods to exploit vulnerabilities for an attacker to gain access in order to inject their attack. These vectors may target various components and subsystems of the ever-growing ITS. Further, an attack vector on one such component/subsystem will tend to lead to effects on other components/subsystems due to the connectivity between them. Then, this work is interested in how those attack vectors may be used to modify the control variables of an ITS application to achieve various objectives depending on attacker profiles, timing, attacker budgets, and attack costs.

IV. Threat Modeling of an ITS

A. Attack Vectors

Many embedded systems within the subsystems of ITS typically have several vulnerabilities that may be exploited remotely once wireless communication is introduced to them, e.g., Dedicated Short-Range Communications (DSRC/IEEE 802.11p), Wireless Access in Vehicular Communications (WAVE/IEEE 1609), cellular (4G, 5G), Bluetooth [19], [21]–[25]. Besides this, attackers can directly alter the software in internal vehicular hardware, such as Electronic Controller Units (ECUs), via the On-Board Diagnostic (OBD) port and infotainment system [26]. Additionally, peripherals such as sensors including induction loop detectors, tire speed (Hall effect) sensors, GPS sensor/GNSS receivers may all be targeted for manipulation or spoofing [16], [25], [27]–[29].

These all may be exploited to perform an attack that will impact the average network flow and waiting time. The attacker may vary from a teenager performing hacks for fun [30], to an angry employee [31], to terrorist organizations. Attacker objectives may vary from slowing down traffic for a single vehicle to macroscopic scale traffic congestion. In this work, we focus on the macroscopic scale; but if desired, the attack models and impact metrics may be configured to focus on an individual vehicle or a smaller subgroup of vehicles.

B. Attaining System State Knowledge

With or without access to the wireless network, the attacker may attain knowledge about the current state of the system,such as the current timing plan configuration, the physical state of the intersection, or vehicle speeds and positions. Without network access, the attacker may easily use off-the-shelf equipment (e.g., RADAR, Infrared, wireless magetometers, acoustic sensors) or their observations because the traffic intersection is in a public space. With access, they may use the measurements from existing vehicle and system sensors. We may also assume that, with the introduction of more technology (i.e., image processing, smarter sensors) it will be even easier to accurately estimate the current system state. Works [13], [24], [25] describe in more detail how these are possible in a realistic setting.

V. System Modeling

As we have mentioned, we have come up with two alternate architectures to implement our attacker and attack models on and to evaluate the usefulness of our impact metrics.

A. Traffic Network Model

In both architectural implementations, we use the Single Ring Road Network Model. The Single Ring Road Network Model is useful to attain the average traffic network performance (whether just a single junction or an entire grid network). In [33], researchers experimented and studied traffic behavior using a single ring road. From their studies, they observed that after certain numbers of vehicles (i.e., called critical densities), the asymptotic traffic behavior changes despite no existing bottleneck in the road. Studies have shown that observed behaviors regarding this model match with empirical studies on actual traffic as well [32]–[34].

As shown in Figure 2, we have also previously designed a real-life experimental setup of the model using robotic cars equipped with sensors and Arduino boards [32]. The logic used by these robots based on the sensors is called the Optimal Velocity Model (OVM) and is similar to car-following logic used in this paper. Despite the sensor inaccuracies, the experimental results from this work positively emulated the

[1] *https://github.com/AICPS/ITSAttackModeling/*

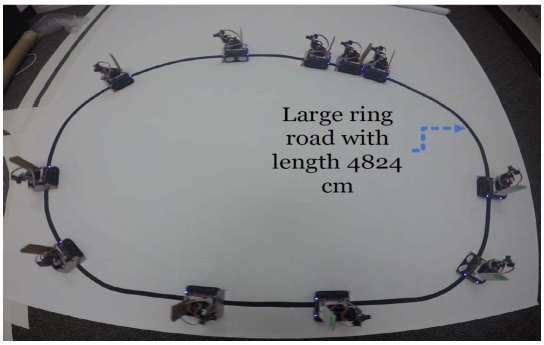

Fig. 2. The test-bed in our previous work [32] for a large Single Ring Road model with length 4824 cm and 11 robotic vehicles.

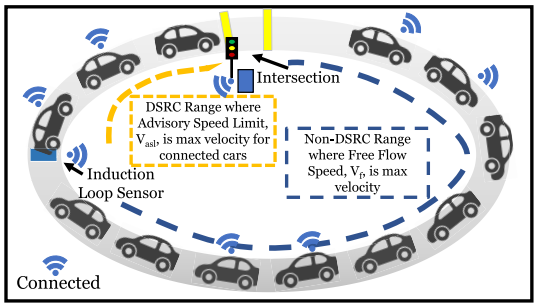

Fig. 3. *Single ring road network model and ASL control system*. Vehicles will receive ASL velocities via V2X (e.g., DSRC) to avoid arriving at the intersection of an actual road during the red time.

empirical and theoretical results from previous studies and may be used for experimentation of car-following models and connected vehicle-based control algorithms.

VI. V2X Advisory Speed Limit (ASL)

In this section, we go over the components of V2X ASL. For some preliminaries, the typical vehicle length is around 5.5m (we use 5m) according to the Highway Manual [35] and the minimal safety headway aka safety cushion distance between a vehicle's head and the front vehicle's tail is approximately 2m [36]. The Jam Distance ($p = 7m$) is the sum of the typical vehicle length (5m) and the safety cushion (2m). The Time Gap (τ) of 1.5s is then derived from the Jam Density ($k_j = 1/p$) and the widely accepted shockwave speed, $w = 4.67m/s$ [37]. Both τ and k_j are calibrated from the NGSIM data set in [38]. The Market Penetration Rate ($MPR \in [0, 1]$) is assumed to be 1.0 for this taxonomy, but it may be adjusted in the V2X ASL algorithm and car following models for future studies. We refer the readers to the variable names and their definitions in Table I, many of which have been derived and studied in previous studies.

A. Vehicles That Have Not Received Advisory Speed Limit

Figure 3 portrays our V2X ASL use case and Single Road Network model with road of length $L = 900m$ between two consecutive intersections. Vehicles arrive within the communication range of a Roadside Unit (RSU) in an intersection at the 600m point in our network model, $L - L_{DSRC} = 900m - 300m = 600m$. This is the first time they are able to begin the ASL procedure. The traffic control system receives a vehicle's Basic Safety Message (BSM) [2]–[4] and uses the vehicle's kinematics and single induction loop (positioned at 600m) data to compute the vehicle's estimated time of arrival to the intersection, $ti(t, n)$. $ti(t, n)$ is based on the number of detected vehicles that are in front, $vif(t, n)$ and the estimated

TABLE I
System Model Variables and Parameters

Cycle Length (T)	60s
Green Time (G)	24 s
Yellow and All-Red Time (Y)	6 s
Red Time (R)	30s
Effective Green Time Ratio (π)	0.5
Simulation Time (t_{sim})	1200 s
Free Flow Velocity (V_f)	15 m/s
Road Length (L)	900 m
Intersection Length (L_{int})	10 m
Length of Road Before Intersection (L_1)	890 m
DSRC Range (L_{DSRC})	300 m
Induction Loop Sensor Position ($L - L_{DSRC}$)	600 m
Jam Distance (ρ)	7 m
Vehicle Length	5 m
Minimum Headway Gap	2 m
Time Gap (τ)	1.5 s
Vehicle Step (Δn)	1
Time Step (Δt)	$\tau\Delta n = 1.5s$
Critical Density 1 ($k_{c,1}$)	15/L veh/m
Critical Density 2 ($k_{c,2}$)	76/L veh/m
Capacity Critical Density (k_c)	31/L veh/m
Capacity Flow of BA-Newell Model ($C = k_c vf$)	.508 veh/s
Capacity Flow of Intelligent Driver Model (C_{IDM})	.4 veh/s
Jam Density (k_j)	1/p=128/L veh/m
Shock Wave Velocity (w)	4.67 m/s
Average Network Flow (q)	0-C
Waiting Time Ratio (r_{wait})	[0,1]
Number of Vehicles (N)	0-$k_j L$
Vehicles in Front (vif)	0-vif_{max}
Maximum Number of Vehicles in Front (vif_{max})	L_{DSRC}/ρ
Estimated Intersection Arrival Time (ti)	[t, t_{sim}]
GPS Error x_{err}	[-5, 5] m
DSRC Delay Mean t_{comm}	$\max(\Delta t, 0.1s)$
Market Penetration Rate (MPR)	1.0

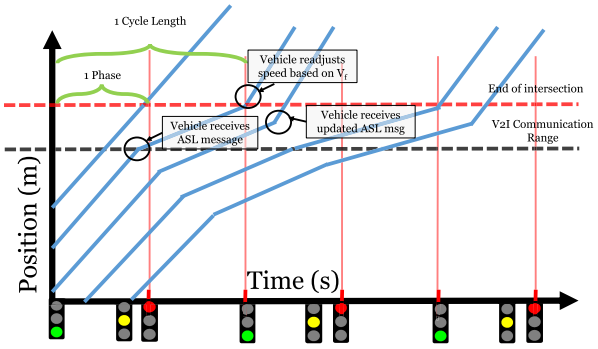

Fig. 4. Representation of vehicles' path traces (each diagonal solid blue line represents the path of a vehicle) when ASL is implemented at a junction (end of intersection is start of ring road in our models). If ASL was not implemented, vehicles would arrive at the red times and cause other vehicles to wait until the light turns green.

arrival time of the vehicle in front $vif(t,n)$. From $ti(t,n)$ and the traffic light schedule, it computes v_{asl} to replace the normal speed limit, v_f, and sends it to the vehicle inside a unique ASL message.

$$ti(t,n) = min(t + \frac{L_1 - \tilde{x}(t,n)}{v_f},\ t + \frac{vif(t,n)}{sr}) \quad (1)$$

where $sr = 1800/3600$ vehicles per second is the typical intersection service rate.

Afterward, v_{asl} may be computed using the expected arrival time (ti) of a vehicle using Equation 2.

$$\begin{aligned} v_{asl}(t,n) &= min(v_f, \frac{L_1 - \tilde{x}(t,n)}{ti(t,n) - t}) \\ where\ \tilde{x}(t,n) &= x(t - t_{comm}, n) + x_{err} \\ &\quad + v(t - t_{comm}, n)\tilde{t}_{comm} \end{aligned} \quad (2)$$

B. Vehicles That Have Received an Advisory Speed Limit

Vehicles that have received their first ASL velocity may/may not need to update their ASL velocity based on the current state of the roads and traffic junction.

If the expected arrival time $ti(t,n)$ is greater than current time step t, the traffic controller will send the vehicle an updated v_{asl} according to the vehicle's positional information in the latest BSM (see Equation 3).

$$v_{n,asl} = min(v_f, \frac{L_1 - x(t,n)}{ti(t,n) - t}) \quad (3)$$

When a vehicle's expected arrival time $ti(t,n)$ is less than or equal to t, we must update its arrival time $ti(t,n)$. First we reset the maximum velocity back to v_f, then re-estimate its new intersection arrival time with it. The new arrival time is computed as follows:

$$ti(t,n) = t + \frac{L_1 - x(t,n)}{v_f} \quad (4)$$

Any time that the newly computed arrival time $ti(t,n)$ might be within the red signal phase, the traffic control system detects it, recomputes it so that it is equal to the start of the next green time, and sends out a new v_{asl} based on it. The formula to do so is shown in Equation 5.

$$ti(t,n) = ti(t,n) + T - mod(ti(t,n), T) \quad (5)$$

where $mod(a,b)$ is the modulus function equivalent to $a\ modulus\ b$.

We want to note that there are various ways that V2X ASL may be implemented and this is only but one of them. In fact, before this paper, there are no other works that have utilized this kind of ASL, which attempts to maximize the usage of V2X to reduce overall waiting time and improve traffic flow.

C. Architecture 1: Matlab and Newell's Car-Following Model With Bounded Acceleration

In Architecture 1, we have written the entire implementation in Matlab and therefore have full control over the car-following model, the network model, the signal settings, and vehicle behaviors. However, because of this, our implementation lacks realistic wireless communication networking (links, packets, etc.) and physical channel modeling that Architecture 2 provides, nor does it consist of detailed sensor definitions (e.g., induction loops) or visualizations.

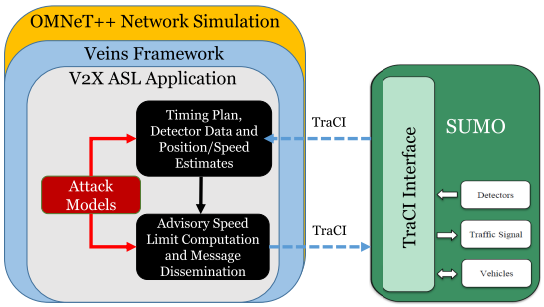

Fig. 5. Architecture 2 comprising of SUMO (car-following and traffic network simulator) and OMNeT++ (communication simulator) connected with Veins Framework through the TraCI (Traffic Controller Interface) API.

In Matlab, we implement Newell's Car-Following Model with Bounded Acceleration (the BA-Newell model), which is simple, has thorough analytical properties, and safety guarantees (unlike more commonly used car-following models) [39], [40]. In this model, vehicles abide by a bounded acceleration but there is no bounded deceleration. Signal settings and model settings are provided in Table I. We consider a 100% aggressive vehicle population where each one decides to go through the intersection if they are able to do so at their current velocity [40].

D. Architecture 2: Veins and Intelligent Driver Car-Following Model

For our second architecture, we use a renowned ITS simulation tool called Veins [41], [42]. Veins is an open-source ITS simulation tool that integrates its own V2X network stack implemented within the communication network simulation tool OMNeT++ [43], [44] with the traffic network, car-following, and vehicle simulation tool SUMO [45]. Via SUMO's Traffic Controller Interface (TraCI), Veins (programmed in OMNeT++) may access various parameters and variables of SUMO. Since the wireless communication considers aspects such as delay and fading in realistic environments and because SUMO is a dedicated and well-developed tool for traffic simulation, Veins is a valuable tool for our purposes to test an V2X ASL use case and our novel attack models. However, in addition to subtle elements that both SUMO and OMNeT++ have (such as randomness in car following and messaging, collision detection and teleportation), it suffers from its incredible complexity and daunting documentation for those outside of the traffic modeling study field.

We designed and implemented a traffic network that emulates both a realistic single junction arterial network and the behavior of the Single Ring Road Network. Because Veins does not include the BA-Newell Model, we instead utilize the popular Intelligent Driver Model (IDM) with $\delta = 32$ (to make it closer to the BA-Newell Model's behaviors) [36]. The IDM capacity ($C_{IDM} \approx 0.4$) differs from that of the Triangular Fundamental Diagram ($C \approx 0.5$), which the BA-Newell Model follows [36], and thus we must normalize the q values

with C_{IDM} and use a 2.5 s/veh saturation headway (aka Service Rate) for the V2X ASL update methods.

VII. Attacker and Attack Modeling Methodology

A. Targeted Components and Control Variables

Considering the ITS components used in V2X ASL, we may narrow down the possible data-based variables that the attackers may target to reduce and reverse V2X ASL performance. We primarily focus on sensor-based attacks but want to make clear that these variables also may be attacked via a tampered OBU or spoofed BSMs.

- *Vehicle to Infrastructure (V2I) Communication Channel*:
- *Induction Loop Sensor*: The variable of interest for this medium is the transmission delay, t_{comm}, leading to the ITS in using an outdated position or velocity.
- *GPS sensor/GNSS receiver*: If the number of vehicles in front of a specific vehicle, vif, is invalid due to physical tampering or packet spoofing, then a vehicle's expected arrival time, $ti(t,n)$, will be adversely affected.
- *Speed/Hall Effect Sensor*: Hall effect sensors are located on vehicle tires that are used to measure speeds. Hence, the velocity value, v, that is computed by the vehicle and transmitted to the traffic control system may be targeted and perturbed.
- *Traffic Controller*: Last, but not least, the v_{asl} itself may be modified through exploitation of a traffic controller itself. Since we can observe the effects of v_{asl} through indirect attacks on other components, we leave it out of this work.

B. Performance Metrics

We primarily target two performance metrics and derive other metrics from them: 1) Average Network Flow and 2) Normalized Waiting Time Ratio. First, we derive these metrics without ASL and without any attack. Then, we derive the impacts that ASL has on these metrics and likewise for various attacks. For both the nominal values and the impacts, we derive another useful metric called the Rate of Change (RoC). The RoC is useful to represent the trend of the system performance as the number of vehicles changes under a given attack (or no attack) scenario. We compare the RoCs before and after the attack and we may observe in the results that the attack models and their derived impacts have common traits across the different system models and simulation tools/settings. This also implies that an RoC trend line may be a useful interpolation function for attack behaviors.

1) Average Network Flow: A Network Fundamental Diagram (NFD) maps an asymptotic average network flow q (product of average network density k and average network velocity $\bar{v}$) with a current traffic state (number of vehicles N or average density k) and is useful in practice for traffic engineers and traffic control systems because they may be quickly used to estimate the current traffic control system performance. To compute an NFD, we must simulate the car-following model and network model for several cycles to compute q corresponding to each defined N, with step $\Delta N = 10\ veh$. A simulation of 1200 seconds is quick, yet sufficient enough, to compute $q = \bar{v}(N/L)$, where $\bar{v}$ is the space-mean velocity for all N vehicles on road of L length. Of notable importance, is how q compares to the capacity, C. And thus, from hereon after, we use q/C to evaluate the traffic flows and have a better understanding of the system performance.

TABLE II

Configurations for Attacker Profile, $\hat{\sigma}$. Note That the Defined Variable Perturbation Ranges ($\hat{\sigma}_{min}$, $\hat{\sigma}_{max}$) for Each Attacker Profile Are Meant to Serve as Guides. They May Be Easily Altered to Fit a Certain System Design or Attack Analysis Objective

Variable / Attacker Profile ($\hat{\sigma}$)	Position $x(t,n)$ $[\hat{\sigma}_{min},\hat{\sigma}_{max}]$	Velocity $v(t,n)$ $[\hat{\sigma}_{min},\hat{\sigma}_{max}]$	Comm. Delay $t_{comm}(t,n)$ $[\hat{\sigma}_{min},\hat{\sigma}_{max}]$	Veh. In Front $vif(t,n)$ $[\hat{\sigma}_{min},\hat{\sigma}_{max}]$	Risk of Detection
Stealthy	$\pm\left[1,\frac{L_{DSRC}}{10}\right]$	$\pm\left[1,\frac{v_f}{5}\right]$	$\left[\begin{matrix}\max(\bar{t}_{comm},\Delta t),\\ \max(2\bar{t}_{comm},2\Delta t)\end{matrix}\right]$	$\pm\left[1,\frac{vif_{max}}{4}\right]$	Low-Medium
Moderate	$\pm\left[\begin{matrix}\frac{L_{DSRC}}{10},\\ \frac{L_{DSRC}}{2}\end{matrix}\right]$	$\pm\left[\frac{v_f}{5},\frac{v_f}{2}\right]$	$\left[\begin{matrix}\max(2\bar{t}_{comm},2\Delta t),\\ \max(5\bar{t}_{comm},5\Delta t)\end{matrix}\right]$	$\pm\left[\frac{vif_{max}}{4},\frac{vif_{max}}{2}\right]$	Medium-High
Extreme	$\pm\left[\begin{matrix}\frac{L_{DSRC}}{2},\\ L_{DSRC}\end{matrix}\right]$	$\pm\left[\frac{v_f}{2},v_f\right]$	$\left[\begin{matrix}\max(5\bar{t}_{comm},5\Delta t),\\ inf\end{matrix}\right]$	$\pm\left[\begin{matrix}\frac{vif_{max}}{2},\\ vif_{max}\end{matrix}\right]$	High
Unrestrained	$\pm\left[\begin{matrix}1,\\ L_{DSRC}\end{matrix}\right]$	$\pm[1,v_f]$	$\pm\left[\begin{matrix}\max(\bar{t}_{comm},\Delta t),\\ inf\end{matrix}\right]$	$\pm[1,vif_{max}]$	Low-High

2) Normalized Average Waiting Time Ratio: We calculate a relative Average Waiting Time Ratio r_{wait}, which is the ratio of average time steps that $v < 0.1m/s$ out of the average trip duration time steps of all vehicles in the network (N), rather than the more frequently used absolute Waiting Time (total time that $v < 0.1m/s$). More uniquely, r_{wait} consists of normalization based on the Average Waiting Time Ratio when there is no ASL, denoted as r_{wait,No_ASL}, of the corresponding architecture. This is so that a fairer comparison between the two architectures may be made.[2] The Average Waiting Time Ratio is $r_{wait} = (t_{wait}/\bar{t}_{wait})/r_{wait,No\_ASL}$ and the impact on the Average Waiting Time Ratio from an attack is $\hat{r}_{wait\,\Delta} = \hat{r}_{wait} - r_{wait}$.

To compute the Waiting Time Attack Impact, we take the difference of the metric before and after attack for the same corresponding number of vehicles N (e.g., $\hat{r}_{wait\,\Delta} = \hat{r}_{wait} - r_{wait}$). We also use the RoC of the normalized waiting time impact as a metric: $\hat{RoC}_{wait\,\Delta} = \frac{\hat{r}_{wait\,\Delta}(N_2) - \hat{r}_{wait\,\Delta}(N_1)}{N_2 - N_1}$.

C. Attacker Profiles

The Attacker Profile function is denoted with $\hat{\sigma}$. Its main purpose is to define the limits to the actual perturbation value $\hat{\alpha}$. In Table II, we provide the characteristics of three different possible attacker profiles: *stealthy*, *moderate*, *extreme*.

[2]When a corresponding waiting time ratio for the no ASL case is less than %10.0, we opt to not use it for normalization and simply take the difference for the impact. Hence values may seem much larger compared to other ones.

A *stealthy* attacker has the objective of keeping their risk of being detected low by subtly injecting attacks with small $\hat{\alpha}$ values that appear to be slightly extreme faults. As expected, such injections may not necessarily create a profound detrimental impact on the system. However, the longer and more frequent the attacks are, the more likely the overall application performance will asymptotically degrade on average. A *moderate* attacker injects riskier attacks, albeit still difficult to detect, to make a dent into the overall performance at a faster rate. An *extreme* attacker has no concern for risk and will (if budget permits) inject a highly noticeable perturbation to their targeted system components/variables. Finally, if desired, an attacker denoted as *unrestrained* may use the full range of the variable perturbation. The attacker profile primarily ensures that the attack perturbation values satisfy the pre-defined constraints of that profile. Due to lack of available space in this work, we assume that there are no constraints on the budget for the attacker.

D. Attack Modeling

The attack models we define in this work are comprised of Attack Values and Attack Timing. Both Attack Values and Attack Timing may be defined as either *static* or *random*. However, Attack Timing also includes additional periodicity elements if desired.

1) Attack Value: In *Static* attacks, the perturbation value remains the same. On the other hand, the *Random* attacks will continuously choose a perturbation value at random (by default, via uniform probability distribution) in the range $[\hat{\sigma}_{min},\ \hat{\sigma}_{max}]$ defined by the Attacker Profile, $\hat{\sigma}$.

2) Attack Timing: We define the attack timing using the following variables: time vector $\overrightarrow{t}$, attack time vector $\overrightarrow{\hat{t}}$, attack frequency $\hat{f}_t$, vehicle vector at current time $\overrightarrow{n}$, vector of targeted vehicles $\overrightarrow{\hat{n}}$, vehicles attacked at current time $\hat{f}_n$, attack time period $\hat{T}$, time between periodic attacks $\hat{\phi}$.

For *Static* timing, the $\hat{f}_t$ may equal $1/\hat{T}$. For *Random* timing, we use the values of $\hat{f}_t$ and $\hat{f}_n$ as probability values for probabilistic events in the time and vehicle domains, respectively. A_t denotes the event that there is an attack during time t, and A_n denotes that there is an attack on vehicle n. Let $P(A_t) = \hat{f}_t$, $P(A_n|A_t) = \hat{f}_n$, $P(\tilde{A}_t) = 1 - \hat{f}_t$ and $P(\tilde{A}_n|A_t) = 1 - \hat{f}_n$. As A_n is dependent on A_t, to know the full probability of an attack on vehicle n at time t (e.g. $P(A_t \cap A_n)$), we must compute $P(A_t)P(A_n|A_t)$. Therefore, $P(A_t \cap A_n) = P(A_t)P(A_n|A_t) = \hat{f}_t\hat{f}_n$ and $P(A_t \cap \tilde{A}_n) = P(A_t)P(\tilde{A}_n|A_t) = \hat{f}_t(1 - \hat{f}_n)$. Note that $P(\tilde{A}_t \cap A_n) = 0.0$ and $P(\tilde{A}_t \cap \tilde{A}_n) = 0.0$.

Having these probabilistic events, we may determine if there will be a perturbation or not at the current time and for the targeted vehicle. This is denoted with the expression: if $(A_t \cap A_n)$, then $\hat{\alpha}(var[i](t,n))$ remains unchanged; else, $\hat{\alpha}(var[i](t,n)) = 0$. Thus, only if the events A_t and A_n are both successful with probability $P(A_t \cap A_n) = f_t f_n$, then the overall value will be a perturbed value, else it will not be. As these events are probabilistic, one may use any kind of probabilistic distribution suitable to their system/attacker/attack model design, and even combine it with defined attacker budgets and attack costs.

TABLE III

CONFIGURATIONS FOR VARIOUS ATTACK MODELS. AN ATTACK MODEL MAY EITHER HAVE STATIC/RANDOM $\hat{\alpha}$ VALUES AND MAY HAVE STATIC/PERIODIC/RANDOM TIMING. NOTE: IF THE ATTACK IS CHOSEN TO BE APERIODIC, THE ATTACK TIME PERIOD WILL BE THE DURATION OF THE ATTACK INSTEAD

Attack Type ($\hat{\alpha}(t,n)$)	Value	Time Domain	Vehicle Domain
Static	*$\hat{\alpha}$ specified by user, but must satisfy: $\hat{\alpha} \in [\hat{\sigma}_{min},\hat{\sigma}_{max}]$*	*Set of Targeted Time Steps: $\vec{\hat{t}}$ Attack Time Period: $\hat{T}$ Attack Time Offset: $\hat{\Phi}$ Attack Start Time: $\hat{t}_0$*	*Set of Targeted Vehicles at Current Time Step t: $\vec{\hat{n}}$*
Random	*Picked using random distribution, e.g. $\hat{\alpha}=U(\hat{\sigma}_{min},\hat{\sigma}_{max})$*	*Probability of Attack at Time Step t: $\hat{f}_t$*	*Probability of Attack on Vehicle n at Current Time Step t: $\hat{f}_n$*

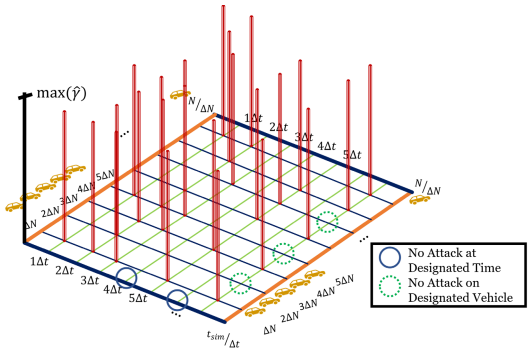

Fig. 6. A visual representation of the combined attacker and attack modeling time and vehicle domain-based function, $\hat{\gamma}$ for an extreme attacker with static attack value and random attack timing.

E. Combined Attacker and Attack Model

Combining the Attacker Profile $\hat{\sigma}$ with Attack Model $\hat{\alpha}$, we may obtain $\hat{\gamma}(var[i],t,n)$ (a visual representation of what the time and vehicle-based function $\hat{\gamma}(var[i],t,n)$ would look like is in Figure 6). After vehicle n sends a message of its kinematics, the final attacked ASL velocity v_{asl} that the RSU will send to the vehicle at time t is defined as the following:

$$\hat{v}_{asl}(t,n) = min(v_f, \frac{L_1 - \hat{x}(t - \hat{t}_{comm}) + \hat{x}_{err} + \hat{v}(t - \hat{t}_{comm})\tilde{t}_{comm}}{(\hat{ti}(t,n) - t)}) \tag{6}$$

where for each variable var, there is a $\hat{var}[i](t,n) = var[i](t,n) + \hat{\gamma}(var[i],t,n)$.

VIII. EXPERIMENTAL RESULTS

In this section, we summarize and provide analyses on some experimental results.[3] For the purposes of data visibility, low simulation time overhead (Veins), and to primarily observe and identify trends, we focus on providing results for each N

[3]We provide videos on several of the Veins attack modeling simulations in *https://sites.google.com/uci.edu/itsattackmodelingresearch/home/v2x-advisory-speed-limit?authuser=0* and our source code on *https://github.com/AICPS/ITSAttackModeling/* to help readers better follow the paper and to serve as tools for the ITS community.

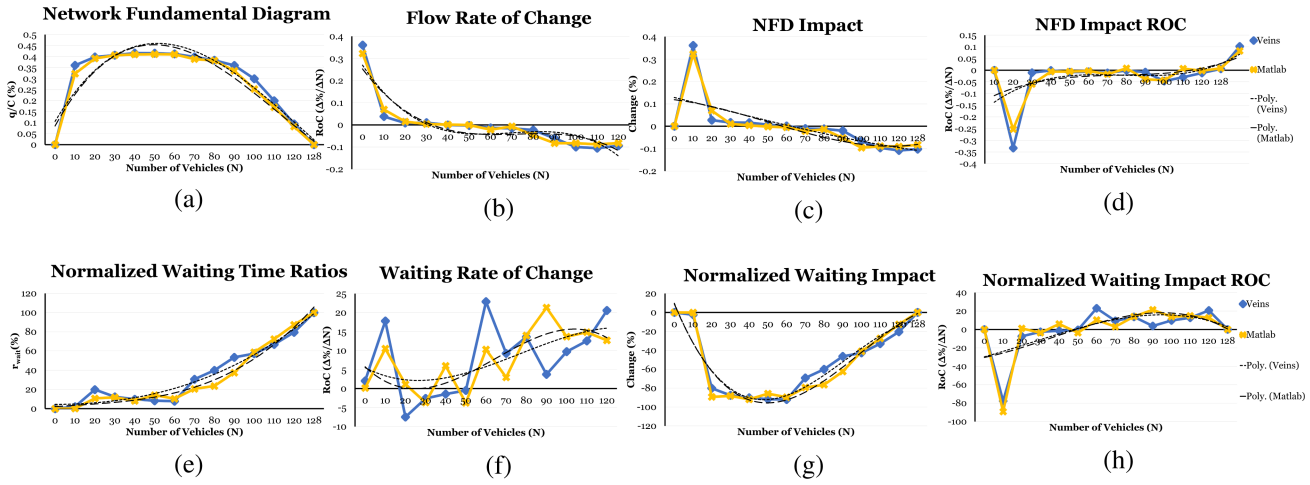

Fig. 7. Simulation results of both architectures for ASL and no attack.

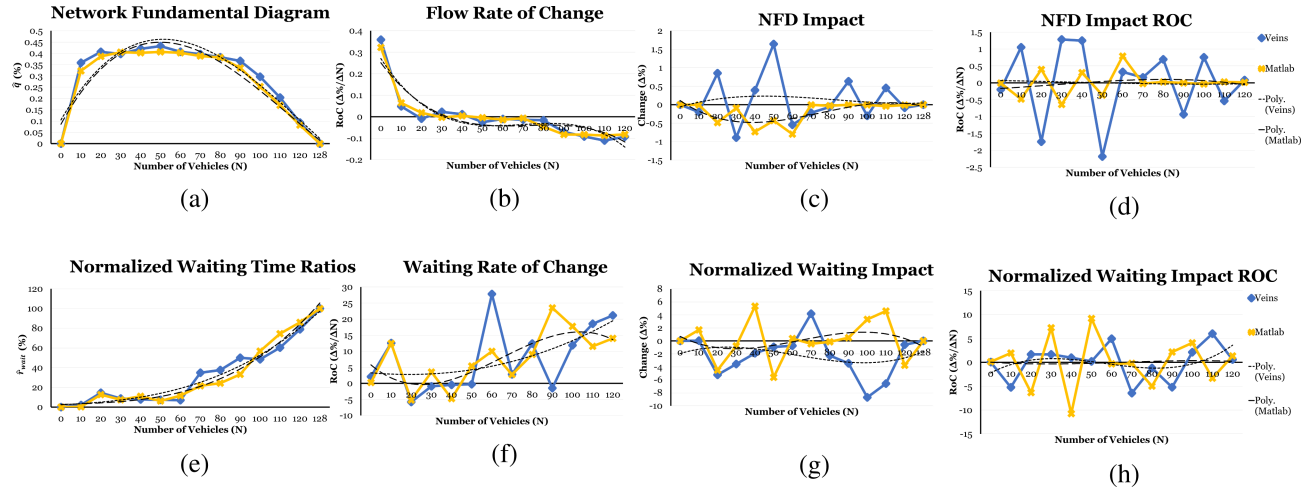

Fig. 8. Simulation results of stealthy attacker, random attack value $\hat{\alpha}(v(t))$, and static aperiodic timing.

in the following set of vehicle numbers: {0, 10, 20, . . . , 110, 120, 128}. Where ΔN is 10 veh except for the last step (8 veh). Our code is readily available and may be used to obtain results for any integer N within [0, 128].

A. Advisory Speed Limit Impacts

Before we discuss the attack-related results, we first analyze the results when ASL is implemented and there is no attack (see Figure 7). From now on we simply denote Architecture 1 metrics with *Matlab* and a yellow line with "x"-like markers and Architecture 2 with *Veins* and a blue line with diamond markers. Notice the similarity between the architectural trend-lines in all the figures. It is interesting to note that ASL acts as a sort of bounded deceleration, which the BA-Newell Model lacks. Hence, although the two car-following models are fundamentally different, the results match up closely when normalized. In contrast, when there is no ASL (not provided), there is up to 30-50% difference in Average Waiting Time Ratios for the region of saturated densities.

B. Attack Impacts

We provide our attack modeling metric results (all in percentage) in Figures 8 to 12 for various types of attacker profiles and attacks. It is important to note that the values should not always be taken at face value; instead, we may infer valuable information from the 3rd degree polynomial trend lines. Through simple observations, the NFD and Waiting graphs (a and e) and their RoC graphs (b and f) are the most similar for the two architectures, and their trend lines match up quite closely. Additionally, although the impact metric graphs (c and g) do not align as well in several cases (mostly due to car-following differences), their RoC counterparts (d and h) do, with average distances across all examples ranging from 1.8-3.5% for for $\hat{RoC}_{q\Delta}$ and 3.3-9.6% for $\hat{RoC}_{wait\Delta}$, and standard deviation ranges, 1.7-3.5% and 3.0-8.7%, respectively. Interestingly, in Figure 12, we observe exceptional similarity between the two architectures across the board. Most likely this is due to the attacker model being an extreme one resulting in more prominent effects.

In addition to the fundamental architectural differences, the discrepancies between the values may be a result of the randomness embedded within the V2X ASL application or within our attack models. There are extreme variations in the range $N \in [20 - 90]$ (some of the largest impacts occur in graphs c and g within each figure) because this is where the network is the most dynamic and vulnerable (i.e., the overall network behavior may range from complete free-flow

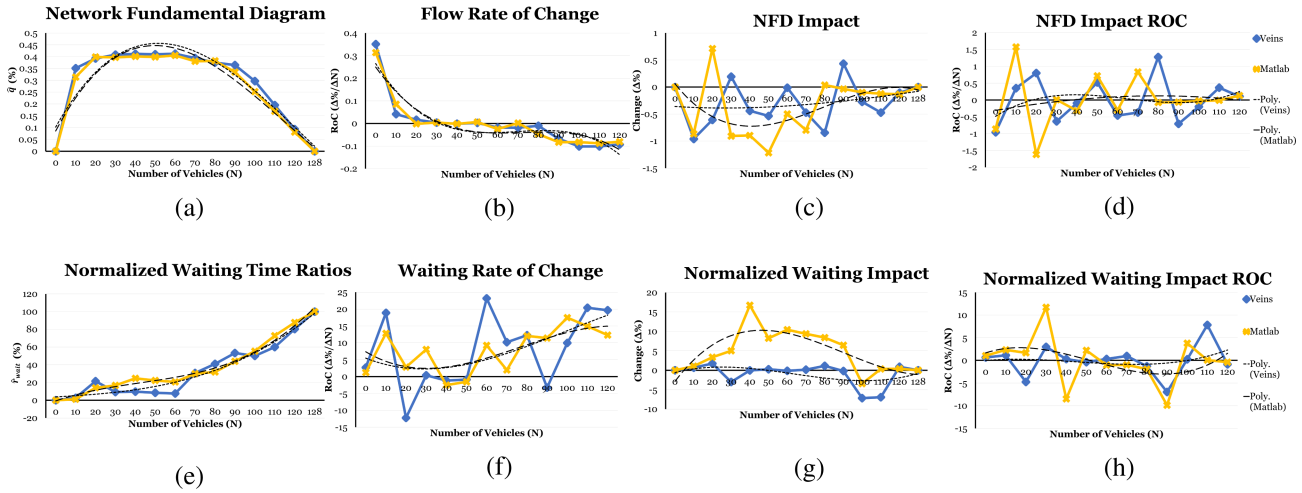

Fig. 9. Simulation results of moderate attacker, random attack value for $\hat{\alpha}(t_{comm})$, and static aperiodic timing for entire simulation, but random vehicle attack probability $\hat{\phi} = 0.4$.

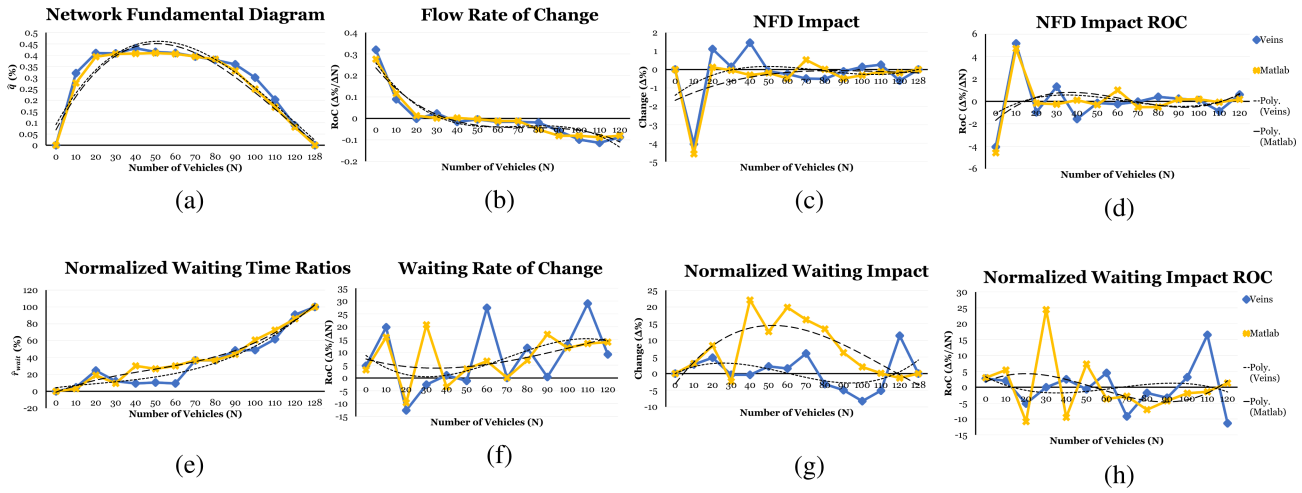

Fig. 10. Simulation results of moderate attacker with static attack value $\hat{\alpha}(x) = 60m$, and static periodic timing of attack period $\hat{T} = 30s$ and offset $\hat{\phi} = 30s$.

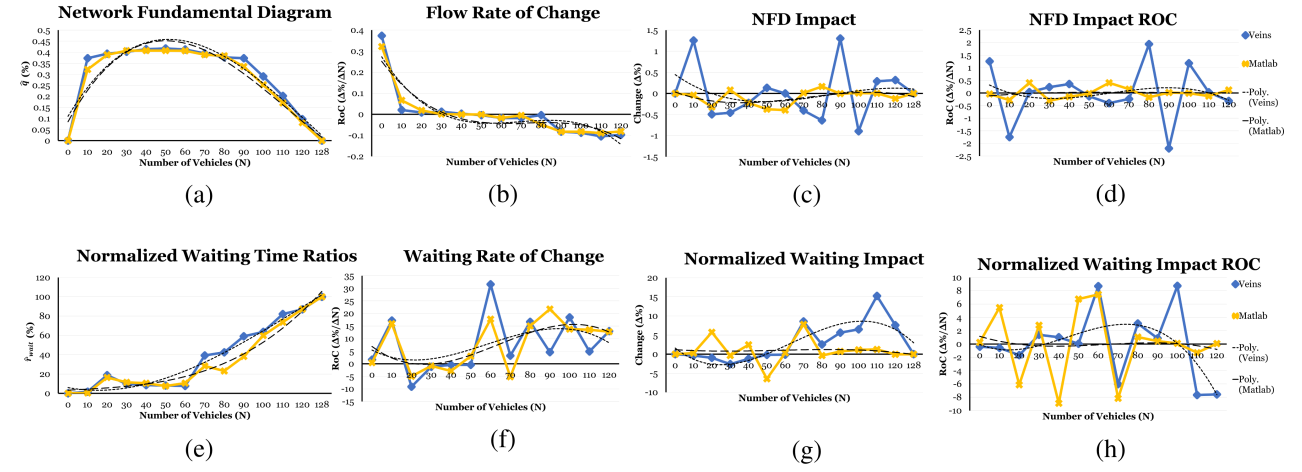

Fig. 11. Simulation results of moderate attacker, attack value $\hat{\alpha}(vif) = -21\ veh$, and aperiodic timing.

to congestion), compared to the network densities on the opposite sides of the spectrum.[4] Further, we observe that the value of $\hat{\gamma}(var[i], t, n)$ may actually incur an opposite effect on the system than what the attacker would have intended (e.g., a positive $\hat{\gamma}$ may force the system to actually help the vehicle arrive earlier during the green time or perhaps help out highly congested traffic densities more than normal ASL). To cope with this, one may program the attack model to dynamically

[4]The impact metric values for some lower densities are not normalized since they would be divided by a less than 1.0 number and therefore appear to be much different than the rest. Nonetheless, since we are more interested in the trends, this does not negatively impact our results.

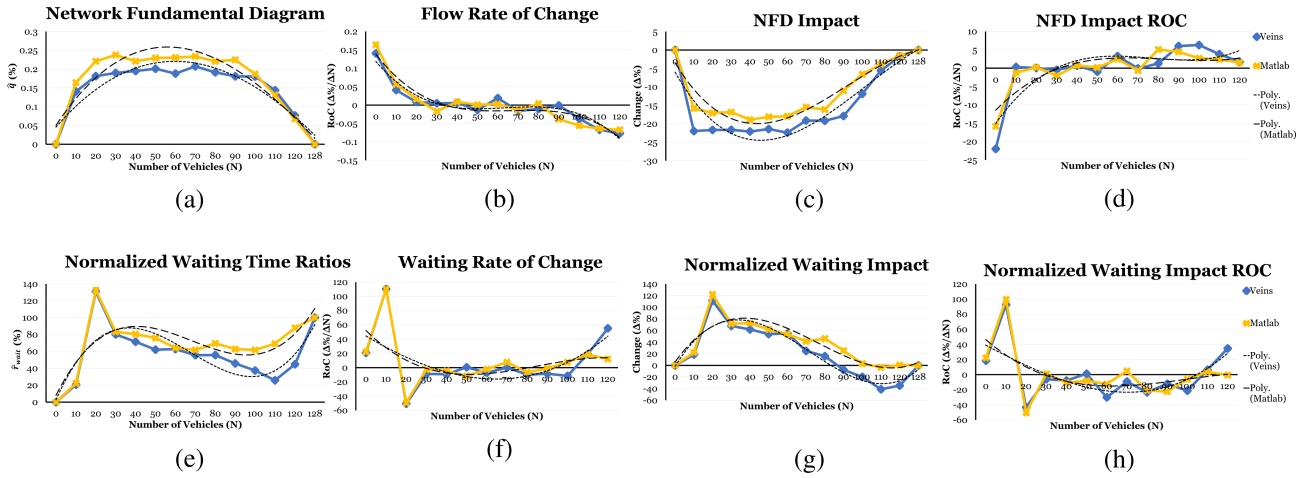

Fig. 12. Simulation results of extreme attacker, random attack value for $\hat{\alpha}(x)$, and random attack time probability $\hat{f}_t = 0.8$, with random attacked vehicle probability $\hat{f}_n = 0.5$.

change the sign of γ or stop the attack on a vehicle or time step to make the impact desirable for an attacker. For example, when the attack may actually help out the vehicle have less waiting time based on the current traffic signal phase or when the network average density is within a certain range (e.g., oversaturated densities where $N > \approx 90$ in Figure 8).

C. Discussion

More intelligent attack models may be constructed from the inferences made from these basic examples. For example, attacks that are permutations and combinations of other ones; attacks that smartly make use of the timing and network state (average density, leader's position, etc.); attacks that incorporate budgets and costs, etc. As the V2X ASL is an ITS use case, other use cases dedicated to improve average speed and average waiting time may also be subject to similar adverse impacts. On the other hand, inferences to improve the security of the system may be made as well. For example, we noted that the saturated densities ($N \in [10, 90]$ in Figure 8) appear to be the most vulnerable. Hence, more resources may be dedicated to protect the ITS when its current average density is in this range. Besides the ITS use case design implementation, other parameters such as minimal advisory speed, aggressiveness, connectivity, and sensor locations may all be studied to improve an ITS' security and performance level as well. Using our methodology, taxonomy, metrics, and tools, all these aforementioned ideas are made possible, especially for those interested in ITS and not from cyber- and system-security related fields.

IX. Conclusion

In this work, we present a methodology and taxonomy to model, simulate, and meaningfully evaluate the exposure and impacts of ITS use cases, such as Advisory Speed Limit control, to attacks. We constructed two distinct architectures and tools to simulate our models and evaluate our metrics. We observed that there is consistency and transferability across the results. Our methodology may serve as a framework for more intricate and interesting attacks to evaluate the security of an ITS design.

Acknowledgment

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. M. Dobersek, "An operational comparison of pre-timed, semi-actuated, and fully actuated interconnected traffic control signal systems," Ph.D. dissertation, UMI Company, Ann Arbor, MI, USA, 1998. [Online]. Available: https://epublications.marquette.edu/dissertations/AAI9912724

[2] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[3] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service ETSI Standard rev. v1.3.2 draft*, ETSI EN 302 637-2 V1.3.2, 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.02_60/en_30263702v010302p.pdf

[4] *IEEE Standard for Information Technology–Local and Metropolitan Area Networks–Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE Standard 802.11p-2010, (Amendment to IEEE Standard 802.11-2007, IEEE Standard 802.11k-2008, IEEE Standard 802.11r-2008, IEEE Standard 802.11y-2008, IEEE Standard 802.11n-2009, and IEEE Standard 802.11w-2009), 2010, pp. 1–51. [Online]. Available: https://ieeexplore.ieee.org/document/5514475, doi: 10.1109/IEEESTD.2010.5514475.

[5] U. S. D. of Transportation. (2018). *About its Standards: Its Standards and the U.S. Dot its Research Initiatives*. [Online]. Available: https://www.standards.its.dot.gov/LearnAboutStandards/ResearchInitiatives

[6] G. A. Ubiergo and W.-L. Jin, "Mobility and environment improvement of signalized networks through vehicle-to-infrastructure (V2I) communications," *Transp. Res. C, Emerg. Technol.*, vol. 68, pp. 70–82, Jul. 2016.

[7] P. Hosseini and K. Savla, "A comparison study between proportionally fair and max pressure controllers for signalized arterial networks," Transp. Res. Board, Washington, DC, USA, Tech. Rep. 16-6738, 2016.

[8] C. Cerrudo and D. Spaniel, "Keeping smart cities smart: Preempting emerging cyber attacks in us cities," *Inst. Crit. Infrastruct. Technol.*, 2015.

[9] M. A. Hasbini, C. Cerrudo, D. Jordan, R. El-Haddahdah, A. Seow, and S. Pawaskar, "The smart city department cyber security role and implications," *Securing Smart Cities*, 2016. [Online]. Available: https://securingsmartcities.org/wp-content/uploads/2016/03/SCD-guidlines_final.pdf and https://securingsmartcities.org/

[10] K. B. Kelarestaghi, M. Foruhandeh, K. Heaslip, and R. Gerdes, "Vehicle security: Risk assessment in transportation," 2018, *arXiv:1804.07381*.

[11] I. Agadakos *et al.*, "Jumping the air gap: Modeling cyber-physical attack paths in the Internet-of-Things," in *Proc. Workshop Cyber-Phys. Syst. Secur. PrivaCy*, New York, NY, USA, Nov. 2017, pp. 37–48, doi: 10.1145/3140241.3140252.
[12] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of things," in *Proc. IEEE World Congr. Services*, Jun. 2015, pp. 21–28.
[13] C. Cerrudo. (2013). *Hacking US (and U.K., Australia, France, etc.) Traffic Control Systems*. [Online]. Available: https://ioactive.com/hacking-us-and-uk-australia-france-etc/
[14] V. L. L. Thing and J. Wu, "Autonomous vehicle security: A taxonomy of attacks and defences," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Dec. 2016, pp. 164–170.
[15] K. Eykholt *et al.*, "Robust physical-world attacks on deep learning models," 2017, *arXiv:1707.08945*.
[16] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR," *Black Hat Eur.*, vol. 11, p. 995, 2015.
[17] A. Lopez, A. V. Malawade, M. A. Al Faruque, S. Boddupalli, and S. Ray, "Security of emergent automotive systems: A tutorial introduction and perspectives on practice," *IEEE Des. Test.*, vol. 36, no. 6, pp. 10–38, Dec. 2019.
[18] Q. A. Chen, Y. Yin, Y. Feng, Z. M. Mao, and H. X. Liu, "Exposing congestion attack on emerging connected vehicle based traffic signal control," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018.
[19] A. Ghafouri, W. Abbas, Y. Vorobeychik, and X. Koutsoukos, "Vulnerability of fixed-time control of signalized intersections to cyber-tampering," in *Proc. Resilience Week (RWS)*, Aug. 2016, pp. 130–135.
[20] A. Laszka, B. Potteiger, Y. Vorobeychik, S. Amin, and X. Koutsoukos, "Vulnerability of transportation networks to traffic-signal tampering," in *Proc. ACM/IEEE 7th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2016, p. 16.
[21] Y. Shoukry, S. Mishra, Z. Luo, and S. Diggavi, "Sybil attack resilient traffic networks: A physics-based trust propagation approach," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 43–54.
[22] J. Reilly, S. Martin, M. Payer, and A. M. Bayen, "Creating complex congestion patterns via multi-objective optimal freeway traffic control with application to cyber-security," *Transp. Res. B, Methodol.*, vol. 91, pp. 366–382, Sep. 2016.
[23] N. Koblitz, "The uneasy relationship between mathematics and cryptography," *Notices AMS*, vol. 54, no. 8, pp. 972–979, 2007.
[24] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, and J. A. Halderman, "Green lights forever: Analyzing the security of traffic infrastructure," in *Proc. USENIX Workshop Offensive Technol. (WOOT)*, vol. 14, 2014, p. 7.
[25] C. Cerrudo, "An emerging US (and world) threat: Cities wide open to cyber attacks," *Securing Smart Cities*, vol. 17, pp. 137–151, 2015. [Online]. Available: https://ioactive.com/pdfs/IOActive_HackingCitiesPaper_CesarCerrudo.pdf
[26] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Secur. Symp.*, 2011, pp. 1–16.
[27] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the spoofing threat: Development of a portable gps civilian spoofer," in *Proc. Radionavigation Lab. Conf. Proc.*, 2008.
[28] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *Proc. Int. Workshop Cryptogr. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2013, pp. 55–72.
[29] D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Drone hack: Spoofing attack demonstration on a civilian unmanned aerial vehicle," Tech. Rep., 2012. [Online]. Available: https://www.gpsworld.com/drone-hack/
[30] J. Leyden, "Polish teen derails tram after hacking train network: Turns city network into hornby set," Tech. Rep., 2008. [Online]. Available: https://www.theregister.com/2008/01/11/tram_hack/
[31] S. Weber, "City sees red after engineers hack traffic signals: Engineers hacked traffic system as part of a labor protest," Tech. Rep., 2009. [Online]. Available: https://www.nbclosangeles.com/news/oddities/city-sees-red-after-engineers-hack-traffic-signals/1866174/
[32] M. J. Carrillo. (2015). *Robotic Cars Test Platform for Connected and Automated Vehicles*. Database Copyright ProQuest LLC; ProQuest Does Claim Copyright Individual Underlying Works; (Mar. 28, 2016). [Online]. Available: https://search.proquest.com/docview/1772400722?accountid=14509
[33] Y. Sugiyama *et al.*, "Traffic jams without bottlenecks—Experimental evidence for the physical mechanism of the formation of a jam," *New J. Phys.*, vol. 10, no. 3, Mar. 2008, Art. no. 033001.
[34] C. F. Daganzo, "The cell transmission model, part II: Network traffic," *Transp. Res. B, Methodol.*, vol. 29, no. 2, pp. 79–93, 1995.
[35] H. C. Manual, "HCM2010," Transp. Res. Board, Nat. Res. Council, Washington, DC, USA, Tech. Rep., 2010, p. 1207. [Online]. Available: https://www.hcm2010.org/system/datas/85/original/Chapter%2031%20-%20Signalized%20Intersections%20Supplemental.pdf
[36] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, no. 2, p. 1805, 2000.
[37] S. Hoogendoorn and V. Knoop, "Traffic flow theory and modelling," *The Transport System and Transport Policy: An Introduction*. 2013, pp. 125–159.
[38] H. Yang, Q. Gan, and W.-L. Jin, "Calibration of a family of car-following models with retarded linear regression methods," in *Proc. 90th Annu. Meeting Transp. Res. Board*, Washington, DC, USA, 2011. [Online]. Available: https://trid.trb.org/view/1093456
[39] W.-L. Jin and Y. Yu, "Performance analysis and signal design for a stationary signalized ring road," 2015, *arXiv:1510.01216*.
[40] W.-L. Jin and J. Laval, "Bounded acceleration traffic flow models: A unified approach," *Transp. Res. B, Methodol.*, vol. 111, pp. 1–18, May 2018.
[41] C. Sommer, "Veins: Vehicles in network simulation," Tech. Rep., 2015. [Online]. Available: https://veins.car2x.org/
[42] C. Sommer *et al.*, "Veins: The open source vehicular network simulation framework," in *Recent Advances in Network Simulation*. New York, NY, USA: Springer, 2019, pp. 215–252.
[43] A. Varga. (2002). *OMNeT++*. IEEE Network Interactive. [Online]. Available: https://www. omnetpp.org
[44] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*. New York, NY, USA: Springer, 2010, pp. 35–59.
[45] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO–simulation of urban mobility: An overview," in *Proc. SIMUL 3rd Int. Conf. Adv. Syst. Simul.*, 2011, pp. 1–6.

Anthony Bahadir Lopez (Member, IEEE) is currently pursuing the Ph.D. degree in computer engineering with the University of California at Irvine, Irvine, CA, USA, under the supervision of Prof. Mohammad Adbullah Al Faruque at the Autonomous and Intelligent Cyber-Physical Systems Laboratory. His research interest includes creating and improving methodologies for intelligent transportation system security. He was a recipient of the prestigious NSF Graduate Research Program Fellowship.

Wen-Long Jin (Member, IEEE) received the B.S. degree in automatic control from the University of Science and Technology of China in 1998, and the Ph.D. degree in applied mathematics from the UC Davis in 2003. He is currently a Professor of civil and environmental engineering at UC Irvine. His research interests include fundamental principles, concepts, models, and methods for analyzing and management of intelligent transportation systems.

Mohammad Adbullah Al Faruque (Senior Member, IEEE) is currently an Associate Professor with the University of California at Irvine in the departments. He is an ACM Senior Member. Among many awards, he was a recipient of the IEEE Technical Committee on Cyber-Physical Systems Early-Career Award 2018, and the IEEE CEDA Ernest S. Kuh Early Career Award 2016.