Cyber-security oriented co-simulation platform for connected and autonomous driving

Ahmed Abdo[1], Guoyuan Wu[2], Nael Abu-Ghazaleh[1]

Abstract— **Connected and autonomous vehicles are a big part of the automotive industry's overall growth trend that may be utilized to improve transportation safety, expand mobility options, lower expenses, and provide new job possibilities. Thus, a complete examination of connected and autonomous driving is required before the large-scale implementation in reality, which may be done affordably and efficiently using a reliable simulation platform. Current traffic simulators ease the operation of connected and autonomous vehicles by offering incremental enhancements to traditional traffic flow modeling approaches, which cannot replicate the features of real-world connected and autonomous vehicles. Moreover, current standard security features that are used by the US Department of Transportation or other research entities are not considered. Network-level evaluation via incorporating both large-scale traffic networks and vehicle-to-anything (V2X) communications is also ignored. This study develops a complete simulation platform for conventional, connected, and automated driving that tightly integrates the main components of V2X communications, traffic networks, and autonomous/conventional vehicle models. Three major open-source components, SUMO, CARLA, and security credential management system (SCMS) based V2X simulator, are integrated and connected via the traffic control interface. The whole simulation platform can be deployed in a Client/Server model. The proposed platform provides an appropriate and trustworthy testbed for examining the possible social/economic effects of connected and autonomous driving under a security credential management system.**

Keywords: **Connected Vehicles, Autonomous vehicles, Cyber-security, co-simulation platform, security credential management system**

I. INTRODUCTION

Nowadays, several transportation challenges need to be addressed with the increasing number of vehicles, including safety, congestion, and air pollution. The connected and automated vehicles' application is a promising field that concerns with these problems [1]. California Partners for Advanced Transportation Technology program (PATH) [2] has shown that the cooperative adaptive cruise control (CACC) system for automated heavy trucks improved the traffic efficiency, throughput and environmental sustainability. University of Michigan researchers [3] demonstrate that both safety and energy efficiency can be significantly improved for both the connected automated vehicles (CAVs) and their neighboring human-driven vehicles. The connected and automated vehicles may bring additional societal benefits by mitigating traffic shockwaves. On the other hand, Forward Collision Warning (FCW) may start to brake after the front vehicle stopped by a few seconds which can be crucial when driving in a high speed or crowded traffic. For the past decade, the U.S. Department of Transportation [4] (USDOT) has been researching and testing a system of vehicles that can sense the environment around them and communicate with other equipped vehicles and infrastructure. These vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications will enable safety, mobility, and environmental advancements that other technologies are unable to provide. They are expected to reduce unimpaired vehicle crashes by 80 percent [5], while also reducing nearly 7 billion extra traveling hours due to traffic. Therefore, a comprehensive assessment of connected and automated driving is essential to study the objectives and goals related to safety, congestion, reliability, environment, freight movement, economic vitality, and other regionally important considerations (e.g., multi-modal travel options, real-time traveler information) need to be framed.

In general, simulators [6] are used to test various aspects of driving including longitudinal/lateral dynamics, driver behavior, and traffic scenarios. There are already many existing simulators developed for specific function verification of connected and automated driving, for example, CARLA for autonomous vehicle (AV) development [7], SUMO for traffic flow modeling [8], and OMNET++ for V2X communication design [9]. Nevertheless, the main problems for the most existing traffic simulators are simplicity [10] and lack of any security standards. This is not suitable for cyber-security related research for CAVs.

First, the simplicity [6] in the current traffic simulators includes simplified representation of CV and not proposing consideration of the real characteristics of CV, especially in terms of context sensing, information communication, intelligent perceptions, and control decisions. Realistic description of mixed traffic flow is still limited in current simulators which consist of both traditional vehicles and CVs with different levels of automation, i.e., the called conventional vehicles, connected human-driven vehicles, autonomous vehicles, and connected and automated vehicles. The lack of such traffic assessments may significantly impede the smooth shifting of CVs from prototypes to industrialized production and wide market adoption. Second, all the existing simulators do not consider Security Credential Management System (SCMS) [11] that is adopted by U.S. Department of Transportation (USDOT) to ensure that connected vehicle technologies operate in a safe, secure, and privacy-protective

[2]Ahmed Abdo and Nael Abu-Ghazaleh are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA. aabdo003@ucr.edu, nael@cs.ucr.edu.

[1]Guoyuan Wu is with the Center for Environmental Research and Technology, University of California, Riverside, CA 92521, USA. gywu@cert.ucr.edu .

manner. For information exchanging among vehicles, roadway infrastructure, traffic management centers, and wireless mobile devices, a security system is needed to ensure that users can trust in the validity of information received from other system users. In this paper, we aim to develop a comprehensive simulation platform for various CAV applications (i.e, platooning, lane merging, intelligent traffic manager, and etc.) from a cyber-physical mobility system perspective, in which the core components of traffic networks and dynamics, autonomous/automated vehicle model, vehicle-to-everything (V2X) communications, and standard protocols for cybersecurity application are seamlessly integrated. Specifically, two popular open-source simulators SUMO, CARLA, along with our V2X simulator are connected via the Traffic Control Interface (TraCI), and the whole co-simulation platform is deployed in a Client/Server mode. Moreover, the architecture design for multi-user access allows multiple human driving simulators to join the same simulation scenario, which may facilitate the exploration of the complicated interactions between conventional vehicles and CVs. The proposed simulation platform provides a realistic traffic environment (e.g. map topology, roadside infrastructure, traffic demands, road users' interaction, and wireless communications), supports different types of vehicle dynamics and driving behaviors including conventional vehicles and CVs. Our main contributions in this paper are as follows.

- We propose an integrated simulation framework for different applications from the cyber-physical mobility systems perspective, in which the core components of traffic networks and dynamics, autonomous and automated vehicle models, as well as V2X communications are tightly coupled.
- A message security solution for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications is realized by using digital certificates issued by the SCMS to authenticate and validate the safety and mobility messages that form the foundation for connected vehicle technologies.
- A case study demonstrates the functionality and feasibility of the simulation platform.

The rest of this paper is organized as follows. A brief literature review is given in Section 2. In Section 3, we discuss the main design elements before generalizing the whole architecture of the proposed simulation framework. In Section 4, we show how the simulation platform works for real-world applications such as Cooperative Adaptive Cruise Control (CACC), with the integrated security credential management system, followed by the conclusion in Section 5.

II. Literature Review

While the realistic implementation and execution of such complex systems entail high costs and major labour resources, simulation is considered a cost effective way for connected and automated vehicle (CAV) maneuver effectiveness verification. In this section, we review those commonly used simulators for CAV modeling and development purpose.

A. Microscopic Traffic Simulators

Generally, a microscopic traffic simulator [6] consists of three major components: (1) transportation network to define road topology at a network level; (2) traffic demand generator to create traffic flow running in the predefined traffic network; and (3) car-following and lane-changing models to regulate individual vehicle driving behavior. In microscopic traffic models [12], vehicles are represented as separate agents, whose motion is governed by specific rules. Those agents may be in interaction, which also has an impact on their behaviors. Some typical simulators include PTV VISSIM [13], AIMSUN [14], and SUMO [8].

VISSIM [12] is a widely used tool for simulating connected and automated vehicles. It provides different interfaces that allow to adapt internal driving parameters: car following model, lane change behavior, operating speed and acceleration. VISSIM models multi-modal realistic transportation operations and creates the best conditions for testing different traffic scenarios before their realization. It [15] has various features, such as traffic flow modeling, traffic light control, vehicle queue length analysis, pedestrian simulation, and also script-based modeling. Script-based modeling is one of the VISSIM's features that are very useful in the development of traffic control algorithms.

AIMSUN [16] is a full-featured simulation model with the ability to obtain detailed state variable information on each vehicle on time scales with better than second-by-second accuracy. AIMSUN has a set of application programming interfaces (APIs) that can well interface with external codes such as EMME/2 and SCATS. AIMSUN's car-following logic has been shown to be realistic and the gap-acceptance behavior is modified based on driver's delay time. Most other models do not represent such phenomena. AIMSUN also has a model for vehicle-actuated NEMA controllers and allows for a look-ahead distance restriction at junctions. Most of the necessary elements are modeled in AIMSUN to support the collection of surrogate measures at a reasonable level of fidelity.

SUMO [17] allows modelling of inter-modal traffic systems including road vehicles, public transit and pedestrians. Included with SUMO is a wealth of supporting tools which handle tasks such as route finding, visualization, network import and emission estimation. SUMO can be enhanced with custom models and provides various APIs to remotely control the simulation via a general traffic control interface (TraCI), which could make it possible to bidirectionally couple traffic simulators with other software.

These simulators provide a general evaluation of traffic dynamics at a large-scale network level such as urban and highway traffic scenarios. However, the vehicle model in these simulators cannot meet the requirement of the high fidelity of vehicle dynamics, e.g., powertrain operation, onboard sensors and wireless communications, at an individual level, which is particularly essential to the connected and automated driving simulation and validation.

B. Game Engine-based Driving Simulators

To model and simulate a real vehicle, all sensors and actuators on-board should be considered. Providing [18] an environment as realistic as possible to simulate conventional vehicles or to test connected and automated vehicles' algorithms (e.g., perception, communication, localization, navigation, and control of the steering and traction system) is required. Thus, options such as 3D rendering, sensor noise, realistic scenario simulation, human-in-the-loop(HuiL), are very important when selecting a physical modeling platform for this type of research.

Modern simulators such as Unity (AirSim), Unreal Engine (Carla, USARSim), ODE (Gazebo, LpzRobots, Marilou, Webots) or PhysX (MRDS, 4DV-Sim), tend to provide realistic 3d rendering which enables high fidelity simulation environment for modeling and evaluation. Standard 3d modeling tools or third party tools can be used to build the environments and dynamic with scripting. C, C++, Perl, Python, Java, URBI, MATLAB or Python are a few of the languages usually used with the most common simulators.

For example, Gazebo [19] is an open source simulator that offers the ability to simulate robot systems in complex situations. It is one of the most popular simulation platforms for the robotic simulation research. Gazebo has a modular design that allows different physics engines to be used, along with high-quality graphics, sensor models, and the creation of 3D worlds and graphical interfaces. Gazebo is built on top of the rendering engine Ogre3D to provide more realistic environments. The use of plugins expands the capabilities of Gazebo to include abilities such as dynamic loading of custom models and the use of stereo cameras, LiDAR, GPS, IMU or RADAR sensors.

Webots is a commercial robot simulator developed by Cyberbotics used in more than 800 universities and research centers worldwide. It has reached a fairly stable state and supports a wide range of hardware. Webots makes use of ODE (Open Dynamics Engine) for the detection of collisions and dynamic simulation of the rigid body. The ODE library allows the physics of the objects to be simulated. Note that the physics plugins can be programmed only in C or C++. The software also provides a large collection of sensors, including a distance sensor, light sensor, cameras, LiDARs, GPS, accelerometer, and force-sensor.

CARLA [7] allows for flexible configuration of the agent's sensor suite. At the time of writing, sensors are limited to RGB cameras and pseudo-sensors that provide ground-truth depth and semantic segmentation. These are illustrated in Figure 2. The user can specify the number of cameras and their types and positions. Camera parameters include 3D location, 3D orientation for the vehicle's coordinate system, a field of view, and depth of field. Our semantic segmentation pseudo-sensor provides 12 semantic classes: road, lane-marking, traffic sign, sidewalk, fence, pole, wall, building, vegetation, vehicle, pedestrian, and others. However, a significant problem with game engine simulators is that when a high-fidelity simulation is required, correct mathematical representation of subsystem in models or software is imperative to achieve realistic calculations [20].

C. Integrated Simulators

Simulation of a transportation network with connected vehicles (CVs) requires a bi-directional coupling mechanism between a transportation simulator and a communication simulator. This mechanism has led to the concept of the closed-loop CV simulator, which has recently drawn a significant amount of research interests within the community. Thus, some simulators extend their capabilities to interact with other software in improving the scalability and fidelity of connected and automated driving simulation. Veins [21] is an open-source model library for OMNeT++, which supports researchers to conduct simulations involving communications of vehicles and infrastructure — either as the main focus of a study or as a component. Veins already includes a full stack of simulation models for investigating vehicles and infrastructure communicating via IEEE 802.11 based technologies in simulations of Vehicular Ad Hoc Networks (VANETs) and Intelligent Transportation Systems (ITS). Plexe [22] is an open source extension to Veins that offers researchers a simulation environment which is able to run experiments in realistic scenarios, taking into account physics and mechanics of the vehicles, communications and networking impairments, and Inter-Vehicle Communication(IVC) protocol stacks. Plexe is easily extensible and already implements protocols to support platooning and cooperative driving applications and several state-of-the-art cruise control models. VENTOS [23] is an advanced version of Veins that supports multi-modal traffic simulation consisting of motor vehicles (passenger cars, buses, trucks, and motor-cycles), bikes and pedestrians. It allows the user to design and verify many dedicated short range communication (DSRC) based applications and monitor the message exchange process between On-Board Units (OBUs) installed on motor vehicles or bikes, Road-Side Units (RSUs) and pedestrians. VENTOS supports all types of V2X wireless communications such as V2V (Vehicle- to-Vehicle), V2I (Vehicle-to-Infrastructure) and V2P (Vehicle-to-Pedestrian), and can be used to design a variety of DSRC based safety/mobility/environment applications such as Emergency Electronic Brake Light (EEBL), CACC vehicle platooning, Multi-hop routing protocols in VANET, and Traffic Signal Control (TSC).

III. Platform structure and design

Mainly, integrated CAV simulators have to include vehicle modeling, traffic dynamics and environment modeling, and V2X communication infrastructure modeling. For vehicle modeling, a vehicle (conventional or V2X based) can be represented as an agent [24] that is characterized by its powertrain and actuators, on-board sensors that provide information about the external environment (such as GPS, radar, camera and LiDAR), communication units for exchanging information with other vehicles to execute and maintain specific maneuvers such as cooperative ramp merging, and the control algorithms (automatically or manually) implemented on each vehicle to guide its operations. A typical connected vehicle (CV) traffic simulator consists of two layers, an entity layer to describe the road users under the constraints of traffic environment, and a cyber layer to model the behaviors of vehicular networks formed by CVs. To simulate the vehicle model, traffic dynamics and environment, and V2X communication infrastructure, we select popular open-source simulators such as CARLA for vehicular physical and control design modeling, and SUMO for generating realistic traffic networks, demands and dynamics, along with V2X simulator (called *V2XSim* and created in python) to produce an integrated simulation platform as shown in Fig. 1. CARLA has been developed from the ground up to support development, training, and validation of autonomous driving systems. In addition to open-source code and protocols, CARLA provides open digital assets (urban layouts, buildings, vehicles) that were created for this purpose and can be used freely. The simulation platform supports flexible specification of sensor suites, environmental conditions, full control of all static and dynamic actors, and maps generation. CARLA include cameras that render the depth of the elements in the field of view in a gray-scale map, provide clear vision of the surroundings that looks like a normal photo of the scene, render elements in the field of view with a specific color according to their tags, and measure changes of brightness intensity asynchronously as an event stream. In addition, it has detection sensors that retrieve collisions between the vehicle and other actors, and detect possible obstacles ahead of the vehicle. It also has GNSS, IMU, LiDAR, Radar, Responsibility Sensitive Safety (RSS), and much more sensors. To build up a 3D traffic environment, OpenStreetMap which is developed by many contributors all over the world, can be used to convert certain sections of the world to an OpenDRIVE format that is compatible with CARLA. OpenSCENARIO is another piece of software that can be used to describe complex, synchronized maneuvers that involve multiple instances of *Entity*. The maneuvers are based on *Actions* synchronized by *Events* defined by *Conditions*. We use SUMO which is an open-source microscopic traffic simulator, for its ability to handle large and complicated road networks at a microscopic (vehicle-level) scale, as well as easily query and control. SUMO provides several car-following and lane-change models to dictate the longitudinal and lateral dynamics of individual vehicles. SUMO and CARLA can be fully synchronized even with two different developed algorithms using the Traffic Control Interface (TraCI). TraCI allows for a real-time control of the simulation and can be used for retrieving values of simulated objects and for manipulating their behaviors. The Vehicle-to-Everything (V2X) communication simulator , which we call *V2XSim*, supports encoding and decoding of ASN.1 DSRC Message Frame in U-PER format. To secure and encrypt basic safety messages (BSM) transmission for each vehicle, Security Credential Management System (SCMS) is used which includes methods of encryption and certificate management to facilitate trusted communications. Authorized system participants use digital certificates issued by the SCMS to authenticate and validate the safety and mobility messages that form the foundation for connected vehicle technologies as defined by IEEE-1609.2. Consequently, a CAV agent model with full functions of sensing, communication, and control is implemented by the combination of CARLA, V2XSim and SUMO. All of the three modules are connected via TraCI. The whole simulation platform can be deployed in a local host or Client/Server mode. The corresponding detailed architecture of the simulation platform is shown in Fig. 2.

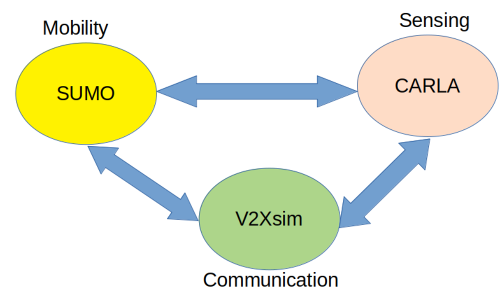


Fig. 1: **Implementation overview of our simulation**

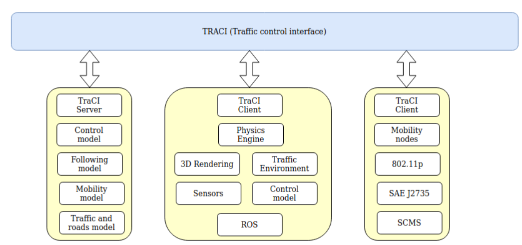


Fig. 2: **System architecture and interface of our simulation platform**

A. Vehicle modeling

Different types of vehicles can be created to model a vehicle in various mixed traffic scenarios, four First, conventional vehicles can be created and defined by SUMO and controlled through different models such as Car-Following Models [25], [26], Lane-Change Models, [27], Traffic Lights [28], etc. Conventional vehicles can be also created by CARLA and fully controlled by human driver via joystick. Vehicles in CARLA follow the second-order kinetic formulas (e.g. speed and acceleration). CARLA is a server/client simulator where the client API is implemented in Python and is responsible for the interaction between the vehicle and the server via sockets. The client sends commands and meta-commands to the server and receives sensor readings in return. Commands control the vehicle and include steering, accelerating, and braking. Meta-commands control the behavior of the server and are used for resetting the simulation, changing the properties of the environment, and modifying the sensor suite. Then, a connected vehicle (CV) in our simulator is modeled as an integration of a conventional vehicle with the communication function implemented by V2XSim using TraCI API commands. The TraCI API commands are split into 13 domains: graphical user interface (GUI), lane, point of interest (poi), simulation, traffic light, vehicle type, edge, induction loop, junction, multi-entry exit, polygon, route, person and vehicle, which correspond to individual modules. We adopt vehicular networking standard, IEEE 802.11p and IEEE 1609.4 DSRC/WAVE. Modeling the V2X communication can be secured by the authentic verification system using SCMS. The SCMS provides the mechanism for devices to exchange information in a trustworthy and private manner using digital certificates. Different vehicle makes and models will be able to talk to each other and exchange trusted data without pre-existing agreements or altering vehicle designs. Each CV contains Pseudonym certificates that are short-term and used primarily for basic safety message authentication and misbehavior reporting. The DSRC system in each CV is simulated to transmit basic safety messages (BSMs) between vehicles and includes information such as exact vehicle location and direction of travel, speed, braking status, and other states so that it could provide safety warnings and trigger actions such as automatic braking or other collision avoidance maneuvers or warnings. The BSM is updated and transmitted 10 times per second. Finally, autonomous vehicles (AVs) and connected and automated vehicles (CAVs) are created and fully controlled by CARLA for the physical dynamic model (the same as the conventional vehicle model). In addition, AVs are equipped with various on-board sensors (e.g. lidar, GPS, camera, etc.) to collect the environment information and CAVs have an extra communication component to represent the (DSRC) on-board units with the implementation in V2XSim. The AVs/CAVs can be controlled using traditional control algorithms such as model predictive control and consensus control, or advanced AI algorithms, e.g., reinforcement learning, and fuzzy control. The co-simulation with SUMO makes for an additional feature. Vehicles can be spawned in CARLA through SUMO, and managed by the later as the Traffic Manager would do. For communication connectivity, V2XSim and SUMO are communicated simultaneously using TraCI APIs.

B. Communication interface

TraCI (short for Traffic Control Interface) gives access to a running road traffic simulation, and allows to retrieve values of simulated objects and to manipulate their behaviors on-line. In our simulation, TraCI supports with timely and seamless information synchronization among SUMO, V2XSim, and CARLA which is very important to implement any maneuver in the simulation platform. TraCI uses a TCP-based client/server architecture to provide access to SUMO. Thereby, SUMO acts as server that can be started with perdefined port to listen on for incoming connections. TraCI can be used to couple multiple processes: A SUMO server process and one or more TraCI client processes (V2XSim and CARLA in this simulation). It executes all commands of a client in sequence. In order to have a predefined execution order, every client should issue a Set-Order command before the first simulation step which assigns a number to the client and commands from different clients during the same simulation step.

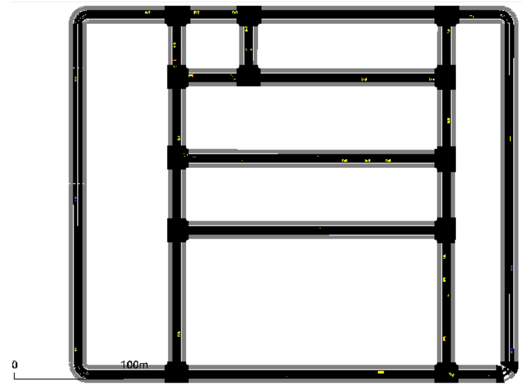


Fig. 3: **A screenshot of SUMO simulator**

C. Information synchronization

V2X is based on information type supported via a periodic beacon, i.e., broadcast message to other CVs to inform the current kinematic states to neighbors. specifically, Basic Safety Messages (BSMs). The BSM is one set of messages defined in the SAE Standard J2735, *Dedicated Short Range Communications (DSRC) Message Set Dictionary*. Each set of messages in the standard, including the BSM, is made up of a set of data frames, which in turn are made up either of other data frames or data elements. The BSM consists of a vehicle's 3D position, speed, direction, acceleration, the message's timestamp, and etc.. SUMO updates the status of each vehicle element by running certain car following model. Then, V2XSim requests all vehicles' information in SUMO and updates their positions in its graphic interface. Based on the obtained latest vehicle information, the CV broadcasts/receives the beacon to/from neighbors via V2X communications within the range, i.e., almost 300 meters for on-board units and 1000 meters for road side units. The V2X information is sent to SUMO for updating the corresponding variables stored in SUMO for associated vehicles. All these steps will be executed within the fixed time duration predefined by the platform. On the other side, CARLA requests all vehicles' status from SUMO at the beginning of simulation by running a synchronous script to spawn vehicles in it and to create the same vehicles' types as in SUMO based on the CARLA blueprint library. After V2XSim implements the V2X communications, CARLA will request and update CVs' or CAVs' status including V2X information from SUMO.

IV. Case study

In this section, we first introduce a simple model in the simulation platform that shows the verification of a CV for true (signed) BSM messages and false ones, then implement a use case to demonstrate how false messages can affect the vehicles speed with and without SCMS authentication methods.

A. SCMS authentication

CVs use short-term pseudonym for the on-board units (OBUs) which can be limited to a particular geographic

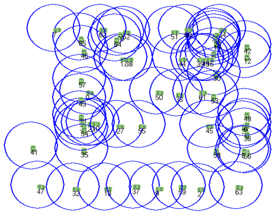


Fig. 4: **A screenshot of V2XSim simulator**

Fig. 5: **A screenshot of CARLA simulator**

region, a particular manufacturer, or a type of devices. These pseudonym certificates encrypt BSM packet generated by each OBU device and broadcast to other CVs. The pseudonym certificates must be changed periodically for privacy reasons. Each pseudonym certificate is valid for a short period of time to protect BSMs from masquerading. As shown in Fig. 6, each connected vehicle uses its digital pseudonym certificates that were issued by an SCMS to authenticate its safety and mobility messages to other connected vehicles and infrastructure. Vehicles broadcast BSMs up to ten times per second to support V2V safety applications. BSMs include the senders' time, position, speed, path history and other relevant information, and are digitally signed. In Fig. 7, the receiver CV evaluates each message, verifies the signature, and then decides whether a warning needs to be displayed to the driver. The correctness and reliability of BSMs are of prime importance as they directly affect the effectiveness of safety applications. To prevent an attacker from inserting false messages, the sending vehicles digitally sign each BSM, and the receiving vehicles verify the signature before taking any associated actions. Once authorized, devices are considered trusted actors in the system. A certification process will ensure that devices meet performance requirements identified across multiple connected vehicle applications and maneuvers and perform as intended. In Fig. 8, a receiver CV is able to validate a false message through the attached sent signature and pseudonym certificate in one of the BSM messages.

B. SCMS effect on Cooperative Adaptive Cruise Control

In the Cooperative Adaptive Cruise Control (CACC), a group of vehicles in close proximity can choose to form a platoon if they are traveling in the same direction at the same time. Once the platoon is created, all vehicles in the platoon travel at the same speed and make lane change decisions as a group. The platoon management protocol refers to the application logic in control of platoon operation.

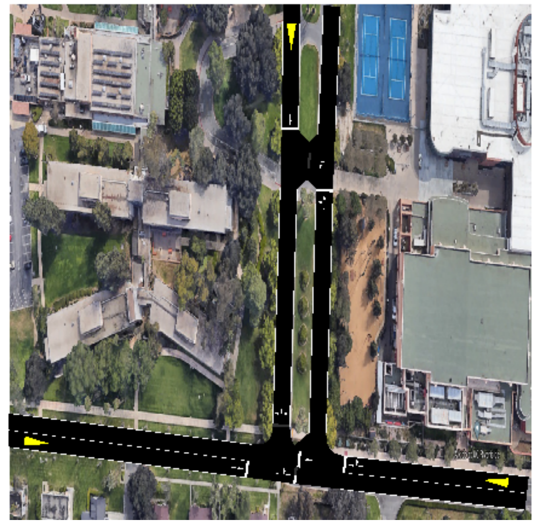
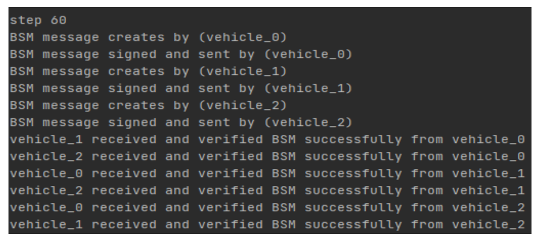
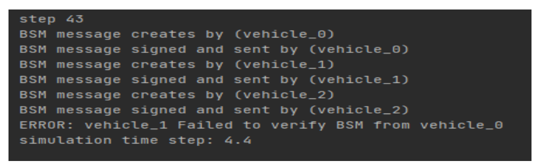


Fig. 6: **A simulated scenario for three connected vehicles using a real map from google earth.**



```
step 60
BSM message creates by (vehicle_0)
BSM message signed and sent by (vehicle_0)
BSM message creates by (vehicle_1)
BSM message signed and sent by (vehicle_1)
BSM message creates by (vehicle_2)
BSM message signed and sent by (vehicle_2)
vehicle_1 received and verified BSM successfully from vehicle_0
vehicle_2 received and verified BSM successfully from vehicle_0
vehicle_0 received and verified BSM successfully from vehicle_1
vehicle_2 received and verified BSM successfully from vehicle_1
vehicle_0 received and verified BSM successfully from vehicle_2
vehicle_1 received and verified BSM successfully from vehicle_2
```

Fig. 7: **A screenshot of signing and sending BSM messages**



```
step 43
BSM message creates by (vehicle_0)
BSM message signed and sent by (vehicle_0)
BSM message creates by (vehicle_1)
BSM message signed and sent by (vehicle_1)
BSM message creates by (vehicle_2)
BSM message signed and sent by (vehicle_2)
ERROR: vehicle_1 Failed to verify BSM from vehicle_0
simulation time step: 4.4
```

Fig. 8: **A screenshot of authentication verification for the received BSM messages**

The leading vehicle acts as the leader/coordinator for the platoon while the rest are followers. The leader makes platoon-wide decisions such as the traveling speed, changing lanes, and merging with other platoons. As the vehicles are controlled cooperatively, they can maintain reduced clearance gaps between each other, allowing for more efficient use of the highway capacity and reduced air drag. Vehicles in the platoon are also able to respond more quickly to changes in traffic compared to ACC; rather than cascading changes with each vehicle responding to its sensors based on the reaction of the vehicle in front of it, the leader can communicate a slow-down (or speed-up) event quickly resulting in faster reaction times. These effects contribute to improving road capacity as well as fuel-efficiency and emission reduction. To form a platoon, the vehicles must be able to communicate with each other as well as to regulate longitudinal gaps. Both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications currently use the Dedicated Short

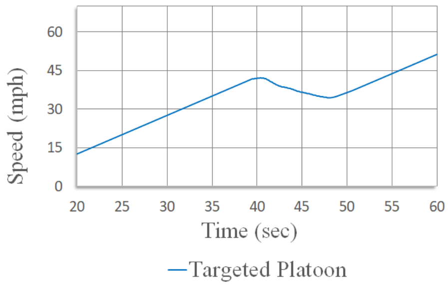


Fig. 9: **Speed profiles without Sybil attack or under SCMS**

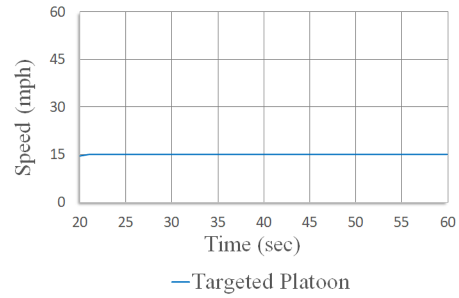


Fig. 10: **Speed profiles under Sybil attack and without SCMS**

Range Communication (DSRC) protocol which has been standardized in IEEE 802.11p. It is also used by the roadside units (RSUs) and on-board units (OBUs) that have transceivers and transponders. RSU triggers vehicles to utilize DSRCs in the OBUs to obtain some essential traffic information such as acceleration, speed, and vehicle position. OBU can send speed and location information through the basic safety message (BSM) and receive messages in certain situations such as approaching an area where speed needs to be lowered. In CACC application, all the connected vehicles send BSMs (ten times per second) that contain some core data elements such as vehicle size, position, speed, heading, acceleration, and brake system status.

In this case scenario, we show the effect of a sybil attack where the attacker tries to create a large number of pseudonymous identities of various fake CVs and uses them to gain a disproportionately large influence. This attacker is one malicious vehicle with the goal of becoming a leader of a specific platoon of some CVs. The attacker pretends to be a 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 to each fake vehicle within 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 which it listens to. The attacker responds to all requests from the rear platoon. This leads to the completion of 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 possible. Fig. 9,10,11,12,13,14 show the platoon speed, time, and fuel changes.

Using SCMS, the attacker is prevented from falsifying messages from another vehicle as each message gets signed with a certificate. Thus, the speed will not be affected. However, SCMS can not prevent a malicious actor from obtaining a certificate and participating in the protocol through replaying the messages (if they are valid), or sending its own messages (with fabricated data) using its certificate.

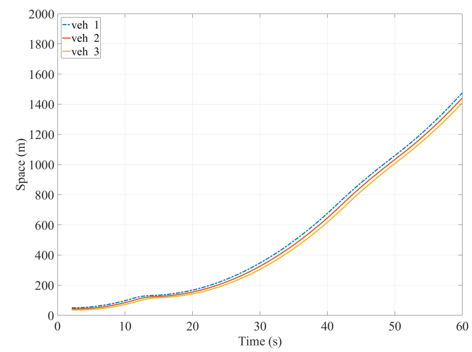


Fig. 11: **Time-Space profiles without Sybil attack or under SCMS**

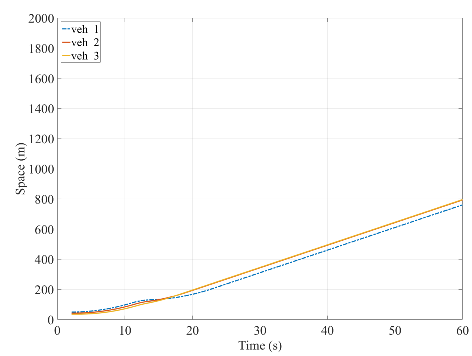


Fig. 12: **Time-Space profiles under Sybil attack and without SCMS**

V. Conclusions and Future Work

In this paper, we design an integrated simulation platform for conventional driving, and connected and automated driving from the cyber-physical mobility system perspective, in which the core components of V2X communications, traffic networks, conventional vehicle model, and connected and automated vehicle (CAV) model are functionally combined. The case study demonstrates the effect of a Sybil attack on the operation of Cooperative Adaptive Cruise Control (CACC) scenarios. Using SCMS, the speed of the platoon will not be affected. It is noted that this simulation platform synchronizes information among three simulators (i.e., SUMO, CARLA, and V2Xsim) via TraCI APIs, which,

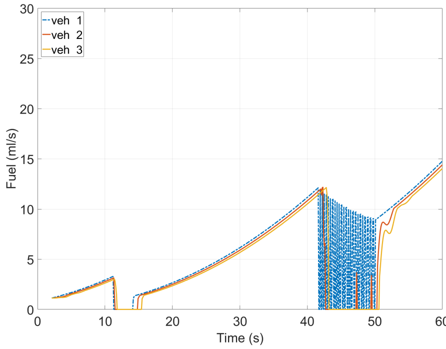


Fig. 13: **Fuel profiles without Sybil attack or under SCMS**

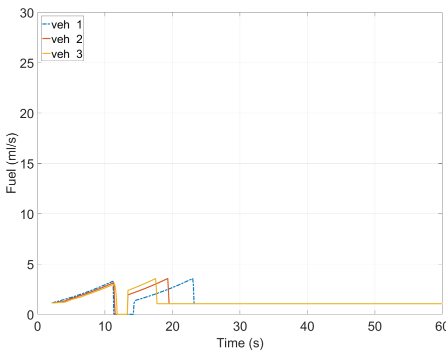


Fig. 14: **Fuel profiles under Sybil attack and without SCMS**

however, may bring heavy communication loads especially in the case of massive CVs. In the future, the usage of the Libsumo functionality may be implemented to reduce the need of socket communication with TraCI. In addition, more security features and attacks' models will be further developed and evaluated with this platform in the future work.

Acknowledgments

This work was partially supported by the University of California Office of the President UC Lab Fees grant number LFR-18-548554.

References

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965856415000804

[2] J. Misener and S. Shladover, "Path investigations in vehicle-roadside cooperation and safety: A foundation for safety and vehicle-infrastructure integration research," 10 2006, pp. 9 – 16.

[3] J. I. Ge, S. S. Avedisov, C. R. He, W. B. Qin, M. Sadeghpour, and G. Orosz, "Experimental validation of connected automated vehicle design among human-driven vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 335–352, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18304443

[4] U. D. of Transportation, "planning for the future of transportation," 2019. [Online]. Available: https://www.its.dot.gov/factsheets/pdf/PlanningFutureTransportation_FactSheet.pdf

[5] intelligent transportation systems, "U.s. department of transportation announces up to 42 million dollars in next generation connected vehicle technologies." [Online]. Available: https://www.its.dot.gov/press/2015/ngv_tech_announcement.htm

[6] D. Jia, J. Sun, A. Sharma, Z. Zheng, and B. Liu, "Integrated simulation platform for conventional, connected and automated driving: A design from cyber-physical systems perspective," 03 2020.

[7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[8] M. Behrisch, L. Bieker-Walz, J. Erdmann, and D. Krajzewicz, "Sumo – simulation of urban mobility: An overview," vol. 2011, 10 2011.

[9] A. Varga, *OMNeT++*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 35–59. [Online]. Available: https://doi.org/10.1007/978-3-642-12331-3_3

[10] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2019.

[11] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A security credential management system for v2x communications," 2018.

[12] P. Gora, C. Katrakazas, A. Drabicki, F. Islam, and P. Ostaszewski, "Microscopic traffic simulation models for connected and automated vehicles (cavs) – state-of-the-art," *Procedia Computer Science*, vol. 170, pp. 474–481, 2020, the 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920305287

[13] M. Fellendorf and P. Vortisch, *Microscopic traffic flow simulator VISSIM*, 06 2011, pp. 63–93.

[14] J. Barcelo and J. Casas, *Dynamic network simulation with AIMSUN*, 01 2005, vol. 31, pp. 57–98.

[15] E. Joelianto, H. Sutarto, and s. antariksa, "Simulation of traffic control using vissimcom interface," *Internetworking Indonesia Journal*, vol. 11, p. 55, 06 2019.

[16] F. H. A. Research and Technology, "Surrogate safety measures from traffic simulation models," 2016.

[17] eclipse.org, "Simulation of urban mobility."

[18] F. Rosique, P. Navarro Lorente, C. Fernandez, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, p. 648, 02 2019.

[19] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," 04 2004, pp. 2149 – 2154 vol.3.

[20] F. Rosique, P. Navarro Lorente, C. Fernandez, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, p. 648, 02 2019.

[21] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, F. Hagenauer, S. Joerer, and M. Segata, *Veins: The Open Source Vehicular Network Simulation Framework*. Cham: Springer International Publishing, 2019, pp. 215–252. [Online]. Available: https://doi.org/10.1007/978-3-030-12842-5_6

[22] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "Plexe: A platooning extension for veins," vol. 2015, 12 2014.

[23] M. Amoozadeh, B. Ching, C.-N. Chuah, and D. Ghosal, "Ventos: Vehicular network open simulator with hardware-in-the-loop support," *Procedia Computer Science*, vol. 151, pp. 61–68, 01 2019.

[24] D. Jia and D. Ngoduy, "Agent-based multiclass model and control of heterogeneous platoon with inter-vehicle communication," 10 2016.

[25] R. Barlovic, L. Santen, A. Schadschneider, and M. Schreckenberg, "Metastable states in cellular automata for traffic flow," *The European Physical Journal B*, vol. 5, pp. 793–800, 10 1998.

[26] A. Kendziorra, P. Wagner, and T. Toledo, "A stochastic car following model," *Transportation Research Procedia*, vol. 15, pp. 198–207, 2016, international Symposium on Enhancing Highway Performance (ISEHP), June 14-16, 2016, Berlin. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146516305464

[27] J. Erdmann, "Lane-changing model in sumo," vol. 24, 05 2014.

[28] D. Krajzewicz, E. Brockfeld, J. Mikat, J. Ringel, C. Feld, W. Tuchscheerer, P. Wagner, and R. Woesler, "Simulation of modern traffic lights control systems using the open source traffic simulation sumo," 06 2005, pp. 299–302.