Platelet : pioneering security and privacy compliant simulation for intelligent transportation systems and V2X

Mathias Kautz, Badis Hammi, Joaquin Garcia-Alfaro

Platelet: Pioneering Security and Privacy Compliant Simulation for Intelligent Transportation Systems and V2X

Mathias KAUTZ*, Badis HAMMI†, Joaquin GARCIA-ALFARO†
*EPITA School of Engineering and Computer Science, France
Mathias.kautz@epita.fr
†SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France
badis.hammi@telecom-sudparis.eu
joaquin.garcia_alfaro@telecom-sudparis.eu

***Abstract*—The development and testing of new applications in Cooperative Intelligent Transportation Systems (C-ITS) environments, which rely on Vehicle-to-Everything (V2X) communication, is frequently supported through simulations. Nevertheless, most of existing simulators are either outdated or do not consider the latest adopted standards. Especially, the security and privacy mechanisms of vehicles and V2X communications. Which leads to incorrect and biased assessments in numerous privacy-aware applications such as Intrusion detection. In this context, we introduce Platelet, which stands as the first V2X simulator compliant with security and privacy standards.**

Source code: https://gitlab.com/Matk3z/platelet
Video: https://www.youtube.com/watch?v=WuIl59mwxi0

I. Introduction

In the context of modern smart cities, Cooperative Intelligent Transportation Systems (C-ITS) represent one of the main use cases that aim to improve citizens' daily life [1]. C-ITS technologies strive to increase road safety, efficiency, and comfort by integrating the processes of sensing, communication, decision-making, and by acting based on the surrounding road environment. A multitude of communication types are intricately involved in a C-ITS environment, commonly referred to as Vehicle-to-Everything (V2X) communications. In the recent years, two main vehicular communication standards have emerged: (1) the Dedicated Short-Range Communications (DSRC) protocol, designed in the US [2], and (2) the Intelligent Transportation System (ITS)-G5 protocol, developed by the European Telecommunications Standards Institute (ETSI) [3]. Both standards are built upon the IEEE 802.11p access layer designed explicitly for vehicular networks. Connected vehicles are also equipped with a cellular interface, which brings numerous advantages. For instance, it addresses the issue of limited coverage of Roadside Units (RSUs) [4] and enables real-time or high-bandwidth consuming applications [5]. Examples of such applications include Vehicle-to-Network (V2N) communications [6] and real-time high-definition video streaming for autonomous vehicles. While many implementations already use the existing Long-Term Evolution (LTE) network, 5G and Beyond 5G technologies demonstrate substantial potential in V2X communications, particularly in the context of autonomous driving [6]. These technologies are expected to be key solutions for enabling efficient autonomous driving [5][7].

Many novel C-ITS applications that are based on V2X communications, undergo initial development and testing through simulation [8]. While industrial entities, such as car manufacturers, often use their proprietary simulators, academia researchers predominantly rely on publicly available simulators (e.g., Veins [9]). In recent years, intrusion detection has emerged as a prominent theme in ITS research, as evidenced by the growing number of papers published in this field over the last years [10]. Most of the intrusion detection works that rely on simulation for validation are founded upon limited or incorrect hypotheses [11]. More precisely, (1) research approaches rely on simulators' output for validation. While a real implementation offers the Intrusion Detection System (IDS) with a complete input of exchanged traffic. This traffic comprise all the established parameters, defined in the standards. Nonetheless, most of the existing simulators only provide a restricted set of outputs and statistics. Consequently, the IDS is confined to the available parameters, deprived of potential benefits from additional parameters that could potentially enhance the detection mechanism. Hence, for precise validations, it is preferable to feed the IDS with all parameters outlined in the standards and enable the IDS to selectively choose the parameters it requires according to its functioning approach. (2) for an accurate and a realistic detection, the IDS needs to consider the security and the privacy mechanisms of the communicating vehicles in the sensed area. Indeed, the security mechanisms in C-ITS environments are already standardized. The standards include the security architectures and the secure message formats. Hence, to handle security requirements, the Public Key Infrastructure (PKI) solution has been adopted by all the standards [1]. The IEEE 1609.2 standard [12] specifies a set of security services to support ITS communications. It defines secure messages formats and processing for Wireless Access Vehicular Environments (WAVE) devices, including methods to secure WAVE management messages and methods

to secure application messages. It also describes administrative functions necessary to support the core security functions. Moreover, the *ETSI ITS Technical Committee Working Group 5* has developed the ITS security architecture, offering security standards and guidance regarding their use [13][14]. Consequently, C-ITS simulators need to consider both the standardized security architecture and the structure of secured messages during simulations. Unfortunately, to the best of our knowledge, none of the existing simulators considers the security structure of messages. Even worse, existing simulators do not even sign messages, although messages signatures verifications could serve as an initial filter at the vehicle level, prior to accepting received packets. (3) Lastly, as vehicles regularly broadcast messages disclosing their position and localization, an attacker can exploit this data to track stations or construct detailed mobility patterns for individual drivers [15]. To counter this privacy issue, a station is provided with a series of pseudonyms, each used for a limited period. More precisely, by relying on the PKI, each vehicle uses two certificates simultaneously: (1) an Enrollment Certificate (EC) and (2) a Pseudonym Certificate (PC) [1]. Known only by the EC Authority (ECA) and its owner (vehicle), the EC is not used in common communications, but rather serves solely for authenticating the vehicle to the PKI in order to request new PCs. In the other hand, the PC is used for the vehicle's communications. To protect the privacy of road users, a regular change of pseudonyms is required. A PC is used by a vehicle for a maximum duration of five minutes, after which a change of the PC is required. Consequently, a vehicle ends up using an extensive number of certificates [16][17]. Two messages sent by the same vehicle but signed with different certificates cannot be linked to the same vehicle by most system actors. Moreover, when a vehicle changes its PC, it changes all its credentials and all the network related information such as Internet Protocol (IP) addresses, Machine (MAC) addresses, Station IDs, and so on. This practice poses significant challenges for IDSs, as they are unable to maintain a continuous trace of a node's activity. Therefore, to simulate realistic scenarios, a simulator must account for the use of the two types of certificates per vehicle (EC and PC), and notably, it must incorporate the periodic PC changes. None of the current simulators considers the certificates' request procedures nor the PC periodic change.

Contributions of this work

In this context, we propose Platelet, the first vehicles network simulator that is compliant with the ETSI security and privacy standards for V2X. Specifically, the contributions of our work are as follows:

- We propose *Platelet* an extension to the *Artery* simulator through the implementation of a security management layer.
- Our simulator is the first to allow vehicles to request certificates, and to regularly change Pseudonym Certificates during a simulation scenario.
- Our simulator is the first to ensure that all the V2X messages sent are correctly signed, and to verify the signatures of the V2X messages received.
- *Platelet* is the first ETSI-compliant simulator that fully implements the security and privacy requirements defined by the standards.
- Our simulator stands as the first to produce an ETSI-compliant packet capture (*Pcap*) for the activity of each vehicle as output.

The remaining of this paper is organized as follows: Section II describes of the most used V2X simulators. Then, Section III introduces our simulator *Platelet* and its architecture. Finally, Section IV concludes the paper and introduces our future works.

II. State of the art

Simulation has always played a crucial role in testing Intelligent Transportation Systems (ITS) for numerous years. In this section, we provide a concise overview of the most used simulators.

A. TraNS

The *Traffic and Network Simulation Environment (TraNS)* [18] stands out as the oldest simulation platform for Vehicular Ad-Hoc Networks (VANET). Built upon the NS2 network simulator and the SUMO traffic simulator, *TraNS* pioneered a realistic simulation approach for VANETs to mitigate the significant discrepancies between simulation results and real-world experiments. Despite its historical significance, *TraNS*, rooted in the outdated NS2 network simulator, faces limitations in supporting large-scale simulations and accurately modeling VANET protocols. The security and privacy mechanisms for vehicles and V2X communications have not been incorporated.

B. GrooveSim

GrooveSim [19] represents another old VANET simulator. It does not rely on existing traffic or network simulators. Instead, its primary objective is to intricately model inter-vehicular communication within a real street map-based topography. *GrooveSim* is designed to comply with the 802.11p/1609 C-ITS communication protocols. It represents a hybrid simulator, facilitating interaction between both real and virtual vehicles in the simulation environment. Nonetheless, the security and privacy mechanisms of vehicles and V2X communications are not implemented.

C. iTETRIS

iTETRIS [20], is an EU-funded simulator. It represents an extension of *TraNS* and uses *SUMO* as a traffic simulator but distinguishes itself by upgrading from NS2 to NS3, a network simulator capable of accurately simulating a large number of nodes. The development of the *iTETRIS* simulation platform was motivated by the lack of simulation platforms capable of precisely modeling and testing C-ITS in expansive scenarios. As an EU project, the *iTETRIS* simulator adheres to the ITS-G5 cooperative ITS communication protocols standard.

TABLE I: Comparison of V2X simulators

Simulator	Traffic simulation	Network simulation	ETSI-compliant	IEEE-compliant	5G and Beyond compatible	Signature implementation	Certificate renewal	pcap logging	EC and PC implementation	Certificate pool implementation
TraNS	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
GrooveSim	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗
iTETRIS	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗
Veins	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗
Artery	✓	✓	✓	✗	✓	✓	✗	✗	✓	✗
Platelet	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓

Like the previously described simulators, *iTETRIS* does not implement security and privacy mechanisms for vehicles and V2X communications.

D. Veins

The *Vehicle In Network Simulation (Veins)* framework [9] represents a contemporary and comprehensive simulation framework built upon *SUMO* and *Omnet++*. *Veins* is designed to be a swift, user-friendly, and highly adaptable platform. It seamlessly incorporates the IEEE 802.11p/1609 cooperative ITS standard, while also offering extensions, such as *Artery*, to support ETSI ITS-G5. *Veins* stands out as the contemporary choice for developing simulations of C-ITS when using the IEEE 1609 and WAVE cooperative standards. Yet, it does not support the security and privacy mechanisms for vehicles and V2X communications.

E. Artery

Artery [21] represents an extension of *Veins*, that is specifically crafted to implement the ETSI ITS-G5 C-ITS communication protocol standard while seamlessly integrating with the *Veins* framework. It relies on *Vanetza* [22], an open-source implementation of the ITS-G5 standard, which incorporates essential features for simulating a network based on the ETSI specifications, including GeoNetworking, and Broadcast Technical Protocol (BTP) protocols. *Artery*, implements few security functions in its source code. Still, no security function is implemented for simulation scenarios.

F. Summary

Numerous VANET simulators exist, and they are highly efficient for various VANET scenarios, making them widely used in academic research. For instance, Veins is the most commonly used simulator due to its comprehensive and efficient integration of traffic and network simulation [23]. However, security and privacy management are critical for all C-ITS simulation scenarios, especially for privacy-aware applications such as intrusion detection. Unfortunately, as highlighted in Table I, none of the most widely used C-ITS simulators implement security management, rendering privacy-aware applications (e.g., IDS) unrealistic and biased. Furthermore, there are very few simulators that implement the ETSI standards compared to those that implement the IEEE standard. *iTetris* and *Artery* are the most effective solutions for simulating ETSI environments. However, they suffer from the same shortcomings described earlier, specifically the lack of consideration for privacy and security requirements. Therefore, there is an urgent need for a tool that supports such simulations.

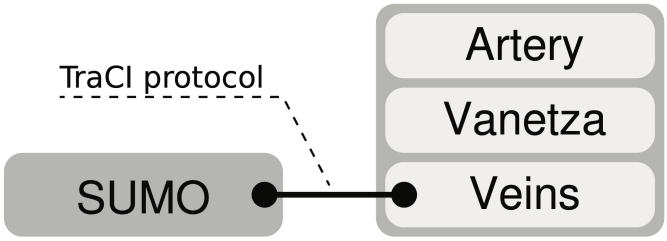

Fig. 1: Components of Artery simulation setup including bi-directional coupling with SUMO [8]

III. Platelet: Toward a Security and Privacy Compliant Simulation

In this section we describe the architecture of Platelet simulator. Platelet is an extension of Artery, hence, we briefly describe the architecture of Artery first. Then, we describe the modules developed for the security management layer for the simulation scenarios.

A. Architecture of Artery

1) General architecture: As Figure 1 shows, Artery is built upon the Veins architecture, consisting of three main components: (1) *SUMO*; a traffic simulator that manages the interactions between cars (e.g., their mobility behavior), (2) *Omnet++*; which manages the communication aspect of the simulation (e.g., physical layer), and (3) *TraCI*; which facilitates bidirectional coupling between them. *Artery* shares the same access layer as *Veins* which is based on the IEEE 802.11p/1609 standard [12]. However, *Artery* diverges from Veins through the use Geonetworking and BTP protocols of the ETSI ITS-G5 standard for the Networking and Transport layer [8]. In *Artery*, the Application and Facilities layers are consolidated into a middleware, with various services (applications) registered within this middleware [8].

Upon the initialization of a new node (e.g., when *SUMO* introduces a new vehicle), the ITS-G5 middleware initializes the node. This process includes the initialization of a GeoNetworking router and the Facilities layer in the node. The middleware also instantiates the node-associated services as

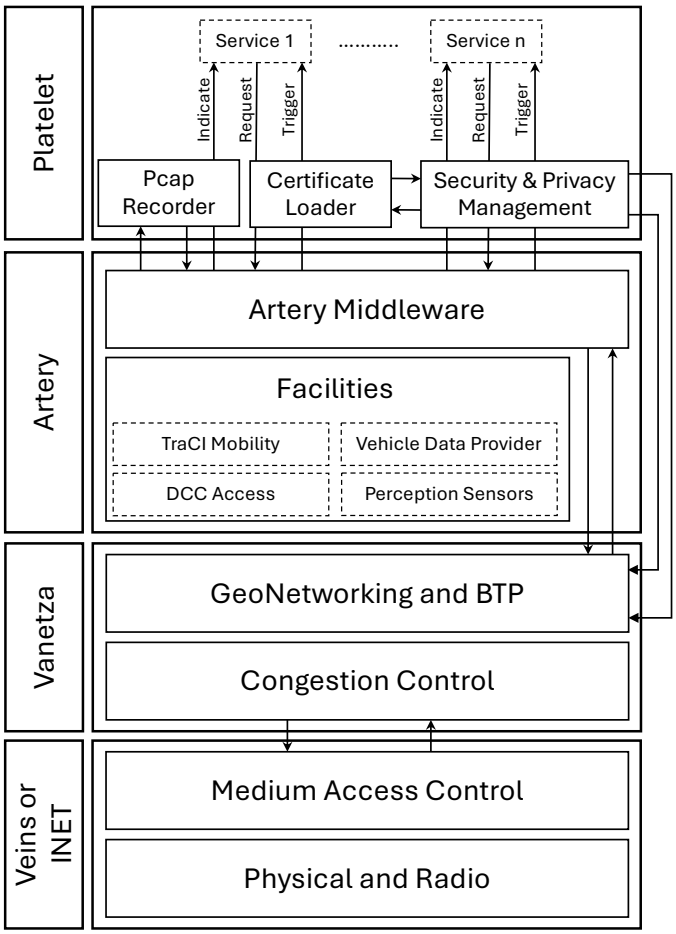

Fig. 2: Platelet architecture

described in the simulation configuration files. These services function as applications for the vehicle, allowing it to send and receive messages (either ASN.1-formatted or *Omnet++* cPackets), gather information from different facilities, and interact with the SUMO node.

2) Services: Within the Artery framework, a service is essentially a class that derives from the *ItsG5BaseService* class. The implementation involves three key methods: *Initialize, Indicate*, and *Trigger*. The *Initialize* method is invoked by the middleware during the simulation's initialization process. The *Indicate* method is called by the middleware when a message is received by the service, and the *Trigger* method is executed by the middleware at specified intervals (configurable in the simulation configuration file).

An illustrative example of a service in Artery is the built-in Cooperative Awareness (CA) service. In the *Trigger* method, this service reads data from the vehicle facilities, uses the information to construct a Cooperative Awareness Message (CAM), and generates a BTP data request. Subsequently, the service dispatches the message using the middleware's request method. In the *Indicate* method, the service verifies the integrity of the received message. If the CAM is deemed valid, it is consumed to update the service's internal data.

B. Platelet

We recall that the required security and privacy features include: (1) the creation of a Public Key Infrastructure (PKI) with at least a Root Certification Authority (Root CA), an Enrollment Certification Authority (ECA), and a Pseudonym Certification Authority (PCA); (2) the ability for vehicles to obtain an Enrollment Certificate (EC) and to request/build a pool of Pseudonym Certificates (PC); (3) the capability for vehicles to change PCs frequently to avoid being tracked; (4) the requirement for vehicles to sign all outgoing messages using PCs (i.e., using the private key associated with the public key of the current PC); 5) the requirement for vehicles to construct outgoing messages with the secured message structure as defined in the ETSI TS 103 097 standard [14]; and (6) The capacity of vehicles to verify the signatures of all the received messages before accepting them.

1) PKI, security and privacy management: The Artery source code includes a module named *SecurityEntity*, which contains some source code for few security primitives. However, these primitives are not fully implemented in the simulation scenarios. More precisely, this module primarily provides functions for encapsulating a *GeoNetPacket* through the *encapsulatePacket* method. For the successful integration of the aforementioned security features within our new framework, the development of new modules was imperative. Initially, we expanded the existing Artery's *SecurityEntity* module to facilitate the instantiation of our newly created modules. To address the issue of the absence of a mechanism to request and then renew certificates, we introduced a *Certificate Provider*. The role of this provider is to supply the necessary number of certificates. Consequently, each node created in the simulation will have a pool of certificates available for cryptographic operations. For instance, each node uses the private key associated with the public key of its PC to compute a signature over the fields of the security header and the message payload. This is done according to ETSI standard specifications and uses the Elliptic Curve Digital Signature Algorithm (ECDSA) as defined by the ETSI TS 103 097 standard [14]. All messages sent during the simulation are signed , and vehicles in the simulation verify the signatures of all received messages. Finally, we have introduced the feature that enables vehicles to frequently change their certificates after a specified period of use. When a vehicle is instantiated, its certificate pool is generated. However, if the user prefers to use their own certificates, this can be achieved through the *Certificate Loader* module. The Figure 2 describes the architecture of platelet.

2) ITS Packet logging : To enhance the simulations outputs, we developed a packet recorder specifically tailored for ITS packets. Similar to the *PcapRecorder* in *Inet*, users can specify the interface they wish to record and the signal they want to monitor (e.g., incoming or outgoing packets, or both). Throughout the simulation, the *PcapItsRecorder* module is initialized on every node and configured to listen to the specified signals. Upon receiving a signal, The module checks whether the message is a *GeoNetPacket*. If it is, the module

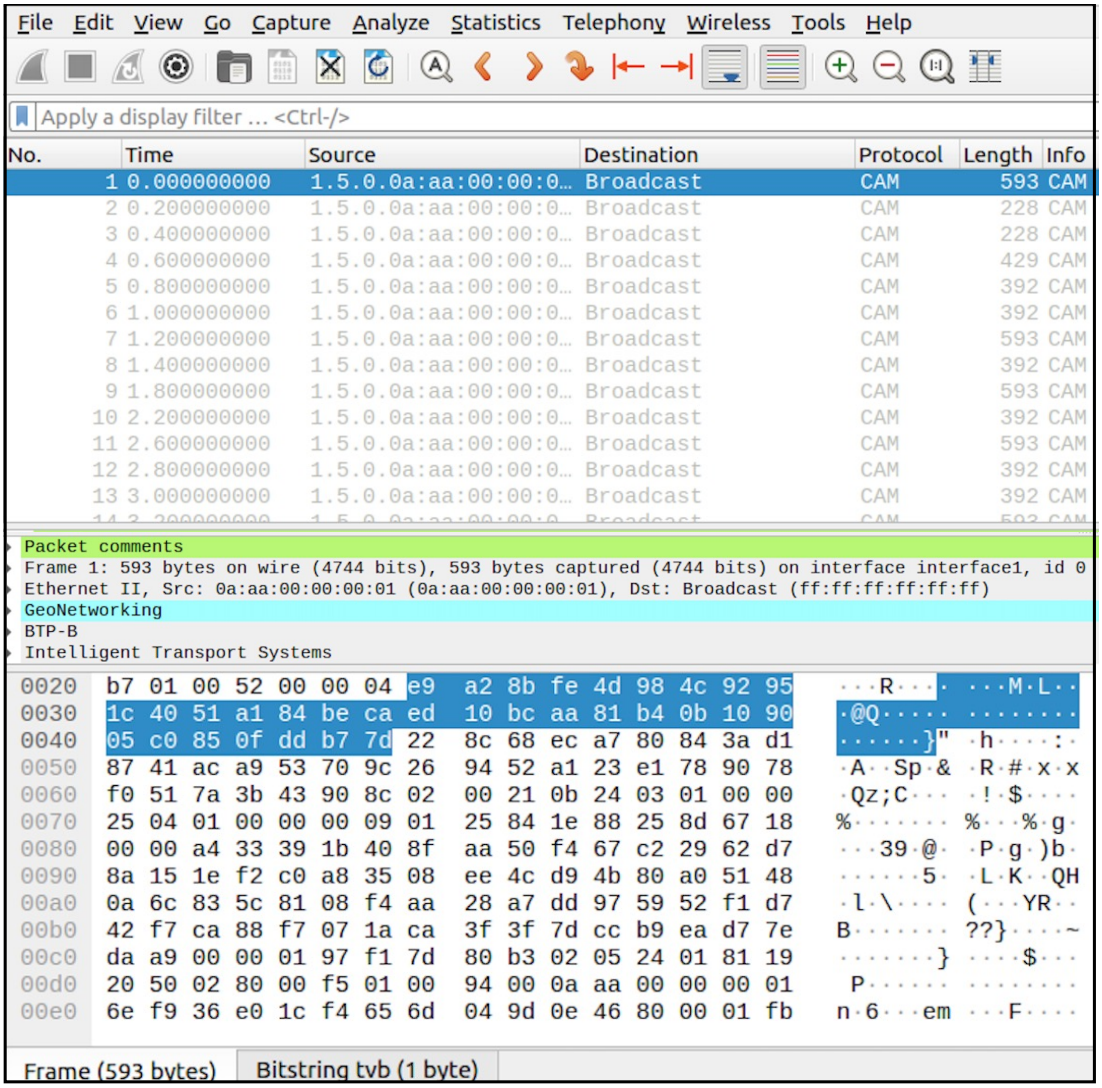

Fig. 3: Wireshark capture of a pcap file obtained from a simulated vehicle's output

serializes various parts of the packet into a *pcap* format and appends it to the *pcap* file. Configurable naming allows for separate files per vehicle, enabling distinct packet captures for each node as it is the case in real implementations. The resulting *pcap* file format is compatible with the ETSI ITS packet dissector in *Wireshark*, enabling visualization of results within *Wireshark* or through the terminal using *Tshark*. Figure 3 displays a *Wireshark* capture of a *pcap* file obtained as output from a simulated vehicle.

Providing *pcap* files as simulation output offers numerous advantages:

- It allows for in-depth packet-level analysis using tools like *Wireshark*. This enables inspection of packet headers, payloads, and metadata to understand the behavior of network protocols and applications.
- It serves as a valuable asset for conducting/developing security-based scenarios, such as intrusion detection, vulnerability assessment, and attack testing.
- It facilitates debugging and troubleshooting. Through the examination of the exact packets exchanged between nodes, users can identify and resolve issues related to protocol implementation, and configuration errors.
- It permits to validate and verify the correctness of network simulations through the comparison of simulated packet exchanges with expected behavior or real-world data ensuring that the simulation accurately represents the intended scenarios.
- It enables the measurement of various performance metrics, such as throughput, latency, jitter, and packet loss. This helps in evaluating the efficiency and effectiveness of network protocols and configurations under different conditions.
- For developers working on new network protocols, *pcap* captures provide a way to observe the protocol in action, facilitating development, testing, and refinement of protocol implementations.

- It contributes to the reproducibility of network simulation experiments. Researchers can share *pcap* files along with their simulation setups, enabling others to replicate and validate the findings.

We provide the source code of Platelet simulator according to the *GNU General Public License (GPL)*. Moreover, we provide a video that describes an example of the simulation of 20 vehicles with *Platelet*. We show how a vehicle obtains a pool of PCs, how messages are signed and how every simulated vehicle provides a *Pcap* output file. Furthermore, *Platelet* is well-suited for simulating attack scenarios. For example, you can watch a demonstration of a Sybil attack scenario using *Platelet* in a companion video.[1] Finally, we have created a Docker container that includes the simulator and all its dependencies. This Docker container provides an easy-to-implement solution, ensuring a streamlined and consistent setup process for users.

IV. Conclusion

In the domain of Cooperative Intelligent Transport Systems (C-ITS), despite the commencement of initial field operational tests, the design and performance evaluation remain predominantly reliant on simulation experiments. These simulations demand a highly specific methodology, as well as specialized tools and models that are not easily found in other domains. Unfortunately, the existing simulators do not consider several essential C-ITS standards, notably disregarding critical security and privacy requirements. This can lead to unrealistic or even biased scenarios. In this context, we proposed *Platelet*, an ETSI-compliant V2X simulator that implements a robust security and privacy management layer. Unlike existing simulators, *Platelet* enables simulated vehicles to: (1) use both Enrollment and Pseudonym certificates; (2) sign outgoing messages; (3) verify the signatures of incoming messages; (4) maintain a pool of certificates and periodically change pseudonym certificates; and (5) generate a comprehensive *Pcap* output detailing each vehicle's entire V2X activity.

To simplify the use of our simulator, we developed a Human-Machine Interface (HMI) that streamlines the simulation setup process for users. This HMI enables control over a wide range of parameters, such as the simulation area, the number of vehicles, the number of messages per second, the number of certificates per pool, and the frequency of pseudonym certificate changes.

This paper presents ongoing work. Our goal is to make *Platelet* a turnkey solution for C-ITS simulation for academia and industry actors, facilitating the development of standards-compliant, security-aware, and privacy-aware applications. Some future operational improvements include the following:

- Provide a comprehensive and holistic technical documentation to explain the simulator's architecture, functionalities, usage, and example scenarios to facilitate users in understanding and utilizing the simulator effectively.
- Enable users to run scenarios with vehicles belonging to different PKIs.
- Develop various attack scenarios that users can easily invoke/trigger during their simulations.
- Evolve the HMI to make it more ergonomic and adaptable to different kinds of scenarios.
- Develop a more complex PKI architecture for simulation scenarios, including a Misbehavior Authority and Linkage Authorities, to enable the simulation and assessment of intrusion detection approaches.
- Extend Platelet to support complement current support to existing standards, including IEEE-based V2X architectures.

In terms of further research, we want to explore the potential of AI in enhancing our simulation platform. Machine learning algorithms can automate scenario design, analyzing historical data to generate dynamic and complex driving scenarios Moreover, the role of AI in enhancing security within C-ITS cannot be overstated [24][25]. AI-driven threat detection and response systems represent a key research direction. Machine learning techniques, particularly those focused on anomaly detection, can be employed to identify malicious behaviors and security breaches. In future iterations of our simulator, we plan to incorporate new AI models and features specifically designed to enhance the design, simulation and testing of AI-based intrusion detection systems . These enhancements will enable more robust and comprehensive evaluation of IDS performance, helping to advance research in securing C-ITS against sophisticated threats.

[1]Companion *Platelet* video available at: https://youtu.be/v9YlUluFh-o

Acknowledgement

Authors acknowledge support from the European Commission (Horizon Europe project AI4CCAM, under grant agreement 101076911).

References

[1] Badis Hammi, Jean-Philippe Monteuuis, and Jonathan Petit. PKIs in C-ITS: Security functions, architectures and projects: A survey. *Vehicular Communications*, 38:100531, 2022.

[2] Khadige Abboud, Hassan Aboubakr Omar, and Weihua Zhuang. Interworking of dsrc and cellular network technologies for v2x communications: A survey. *IEEE transactions on vehicular technology*, 65(12):9457–9470, 2016.

[3] Ribal F Atallah, Maurice J Khabbaz, and Chadi M Assi. Vehicular networking: A survey on spectrum access technologies and persisting challenges. *Vehicular Communications*, 2(3):125–149, 2015.

[4] M Shahid Anwer and Chris Guy. A survey of vanet technologies. *Journal of Emerging Trends in Computing and Information Sciences*, 5(9):661–671, 2014.

[5] Saqib Hakak, Thippa Reddy Gadekallu, Praveen Kumar Reddy Maddikunta, Swarna Priya Ramu, M Parimala, Chamitha De Alwis, and Madhusanka Liyanage. Autonomous vehicles in 5g and beyond: A survey. *Vehicular Communications*, 39:100551, 2023.

[6] Mario H Castañeda Garcia, Alejandro Molina-Galan, Mate Boban, Javier Gozalvez, Baldomero Coll-Perales, Taylan Şahin, and Apostolos Kousaridas. A tutorial on 5g nr v2x communications. *IEEE Communications Surveys & Tutorials*, 23(3):1972–2026, 2021.

[7] Hamidreza Bagheri, Md Noor-A-Rahim, Zilong Liu, Haeyoung Lee, Dirk Pesch, Klaus Moessner, and Pei Xiao. 5g nr-v2x: Toward connected and cooperative autonomous driving. *IEEE Communications Standards Magazine*, 5(1):48–54, 2021.

[8] Raphael Riebl, Hendrik-Jörn Günther, Christian Facchi, and Lars Wolf. Artery: Extending veins for vanet applications. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 450–456. IEEE, 2015.

[9] Christoph Sommer, David Eckhoff, Alexander Brummer, Dominik S Buse, Florian Hagenauer, Stefan Joerer, and Michele Segata. Veins: The open source vehicular network simulation framework. *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*, pages 215–252, 2019.

[10] Brooke Lampe and Weizhi Meng. Intrusion detection in the automotive domain: A comprehensive review. *IEEE Communications Surveys & Tutorials*, 2023.

[11] Badis Hammi, Yacine Mohamed Idir, Sherali Zeadally, Rida Khatoun, and Jamel Nebhen. Is it really easy to detect sybil attacks in c-its environments: a position paper. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):18273–18287, 2022.

[12] Intelligent Transportation Systems Committee & others. IEEE Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages. *IEEE Vehicular Technology Society Standard*, 1609.2:1–884, January 2016.

[13] ETSI TS 102 941: Intelligent Transport Systems (ITS); Trust and Privacy Management. *Technical specification, European Telecommunications Standards Institute*, page 71, May 2018.

[14] ETSI TS 103 097 V2.1.1: Intelligent Transport Systems (ITS), Security header and certificate formats; Release 2. page 22, October 2021.

[15] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. Pseudonym schemes in vehicular networks: A survey. *IEEE communications surveys & tutorials*, 17(1):228–255, 2015.

[16] Benedikt Brecht, Dean Therriault, André Weimerskirch, William Whyte, Virendra Kumar, Thorsten Hehn, and Roy Goudy. A security credential management system for v2x communications. *IEEE Transactions on Intelligent Transportation Systems*, 19(12):3850–3871, 2018.

[17] Virendra Kumar. Special Cryptographic Primitives in SCMS. SCP1: Butterfly Keys. https://wiki.campllc.org/display/SCP, March 2017.

[18] Michal Piorkowski, Maxim Raya, A Lezama Lugo, Panagiotis Papadimitratos, Matthias Grossglauser, and J-P Hubaux. Trans: realistic joint traffic and network simulator for vanets. *ACM SIGMOBILE mobile computing and communications review*, 12(1):31–33, 2008.

[19] Rahul Mangharam, Daniel S Weller, Daniel D Stancil, Ragunathan Rajkumar, and Jayendra S Parikh. Groovesim: a topography-accurate simulator for geographic routing in vehicular networks. In *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 59–68, 2005.

[20] Vineet Kumar, Lan Lin, Daniel Krajzewicz, Fatma Hrizi, Oscar Martinez, Javier Gozalvez, and Ramon Bauza. itetris: Adaptation of its technologies for large scale integrated simulation. In *2010 IEEE 71st Vehicular Technology Conference*, pages 1–5. IEEE, 2010.

[21] Raphael Riebl, Christina Obermaier, and Hendrik-Jörn Günther. Artery: Large scale simulation environment for its applications. *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*, pages 365–406, 2019.

[22] Raphael Riebl, Christina Obermaier, Stefan Neumeier, and Christian Facchi. Vanetza: Boosting research on inter-vehicle communication. *Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)*, pages 37–40, 2017.

[23] Christoph Sommer, Jérôme Härri, Fatma Hrizi, Björn Schünemann, and Falko Dressler. Simulation tools and techniques for vehicular communications and applications. *Vehicular ad hoc Networks: standards, solutions, and research*, pages 365–392, 2015.

[24] Wasim A Ali, Michele Roccotelli, Gennaro Boggia, and Maria Pia Fanti. Intrusion detection system for vehicular ad hoc network attacks based on machine learning techniques. *Information Security Journal: A Global Perspective*, pages 1–19, 2024.

[25] Sparsh Sharma and Ajay Kaul. A survey on intrusion detection systems and honeypot based proactive security mechanisms in vanets and vanet cloud. *Vehicular communications*, 12:138–164, 2018.