Misbehavior Detection System for Cellular Vehicle to Everything Technologies

Michael Aliberti, Max Ellsworth, Jason Inirio, Sam Krasnoff, Julia Zeng, Yixiu Zhu

Executive Summary (Authored by Michael Aliberti)— Vehicle to Everything Technology enables cars and IOT devices with an LTE connection to broadcast positional information such that vehicles can perceive their surroundings without line of sight. Before it is widely adopted, however, its reliability and security must be verified. To this end, we are developing a monitoring system that listens on Vehicle to Everything frequency blocks, verifies message integrity, and leverages machine learning to detect denial of service attacks. This system will consist of a dedicated radio and a web application to display information about frequency block resource usage. We will also be modeling examples of ordinary communication and denial of service attacks to provide data to the monitor. The goal of this project is to provide insight into the strengths and vulnerabilities of the Vehicle to Everything protocol.

Index Terms— Distributed Networks, Machine Learning, Network Protocols, Vehicle Operation

- *Michael Aliberti is with the Department of Computer and Electrical Engineering, Boston University, Boston, MA 02215. E-mail: mjali@bu.edu.*
- *Max Ellsworth is with the Department of Computer and Electrical Engineering, Boston University, Boston, MA 02215. E-mail: maxell@bu.edu.*
- *Jason Inirio is with the Department of Computer and Electrical Engineering, Boston University, Boston, MA 02215. E-mail: jasoninri@bu.edu.*
- *Sam Krasnoff is with the Department of Computer and Electrical Engineering, Boston University, Boston, MA 02215. E-mail: krasnoff@bu.edu.*
- *Julia Zeng is with the Department of Computer and Electrical Engineering, Boston University, Boston, MA 02215. E-mail: zjulia@bu.edu.*
- *Yixiu Zhu is with the Department of Computer and Electrical Engineering, Boston University, Boston, MA 02215. E-mail: zhuyixiu@bu.edu.*

1 Introduction (Authored by Samuel Krasnoff)

As the information age continues to push the boundaries of interconnectivity, vehicular communication is beginning to incorporate modern innovations. Vehicle-to-Everything, or V2X, technology is poised to revolutionize the way that cars interact with their surroundings. By leveraging LTE signals, the V2X protocol allows for information to be exchanged between cars and any wirelessly connectable device, which includes civilian smartphones, bikes, and even other cars. By creating this network of devices, vehicles will be notified of road conditions, accidents, and other unexpected events with unprecedented speed and accuracy. Estimates predict that signals pertaining to left-turn warnings and blind-spot detection alone could prevent upwards of 600 thousand crashes and save over 1,000 lives each year.

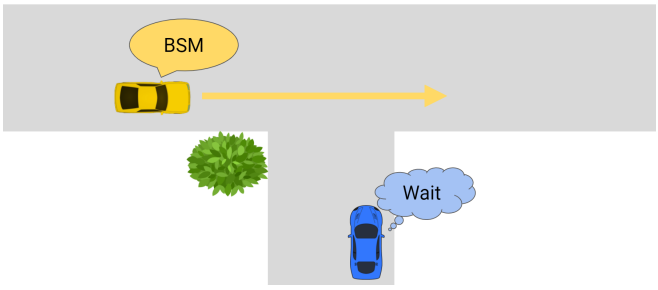


Fig. 1. Within the V2X protocol, cars broadcast Basic Safety Messages (BSMs) which convey location and velocity. This information facilitates inter-vehicle awareness without line of sight.

However, with the arrival of this new technology comes new challenges and malicious actors. In normal use, V2X-capable devices will be transmitting and receiving vital data, like GPS position and velocity. If the transmissibility of messages is poor or if an attacker is able to jam the network through a Denial-of-Service (DoS) attack, then vehicles will be rendered unable to properly assess their surroundings. This compromises the integrity of the mesh network, and the likelihood of collisions skyrockets.

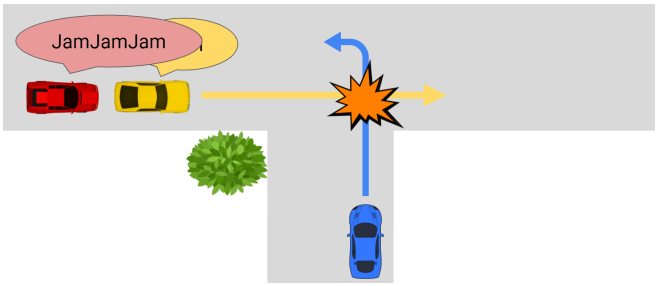


Fig. 2. When an attacker disrupts the transmission of basic safety messages, V2X provides limited information and collisions are far more likely to occur.

This project aims to not only examine the speed and efficacy of a cellular V2X network, but also to explore the detection and possible avoidance of malicious signals sent by attackers. Through machine learning and cyber security principles, a monitoring system will be set up to detect and warn users in real-time about ongoing DoS attacks. By verifying the reliability and security of V2X technology, we can hopefully reap its benefits while avoiding its pitfalls.

1.1 Problem, Purpose, and General Approach

The purpose of this project is to build a monitoring infrastructure to observe communication (e.g. channel resource usage) over a C-V2X network and detect potential anomalies, including DoS attacks. There are four main components to the project as outlined by our customer: (1) demonstrate the ability of transmitted and receiving C-V2X messages, (2) visualize resource usage via a web-accessible API, (3) implement DoS attacks introduced in [2], and (4) design and validate a misbehavior detection system to

thwart such attacks.

To gauge reliability and end-user safety in cellular Vehicle-to-Everything communication, a functional model must be created. This model will include two radios functioning as ordinary vehicles. Normal basic safety messages will be sent on multiple orthogonal frequency bands in the V2X range between these two radios, serving as a source of control data and as a testbed for attack models. A third radio will act as an attacker or team of attackers, attempting to prevent normal operation and cause instability through DoS attacks. A fourth radio will monitor the entire V2X frequency spectrum and pass the data it receives to a connected device, which will in turn store the data in a database. This data can then be visualized and used to train a machine learning algorithm in order to provide insights into V2X resource utilization and attack patterns.

1.2 Deliverables

Before implementing anything in hardware, this project demands two literature reviews. The first is intended to investigate the nature of V2X communication and its execution on software defined radios. The other review will focus on the nature of possible attacks on V2X and how they have been detected in the past, thus informing the design of our monitoring system.

The functional hardware implementation of V2X behavior detailed above serves as a deliverable in-and-of-itself, as it may be used independently to explore other practical dimensions of the protocol. Three distinct DoS attack scenarios - oblivious, smart, and coordinated attacks [2] - must also be modeled in hardware, with documentation of all code changed in the V2X library for ease of replication. The final hardware-adjacent deliverable is a monitoring system to record data across all V2X frequency bands and store said data in a web-accessible API. There must also be a simple GUI for visualization of the monitored data and the overall resource usage of the V2X frequency band.

At the very core of the project is the misbehavior detection system itself. This must be a piece of software running on the monitor which uses data generated by the model to ascertain whether or not a specific snapshot of V2X traffic represents an attack. This will likely take the form of a machine learning algorithm which uses classification techniques to identify normal behavior as well as the three different avenues of attack.

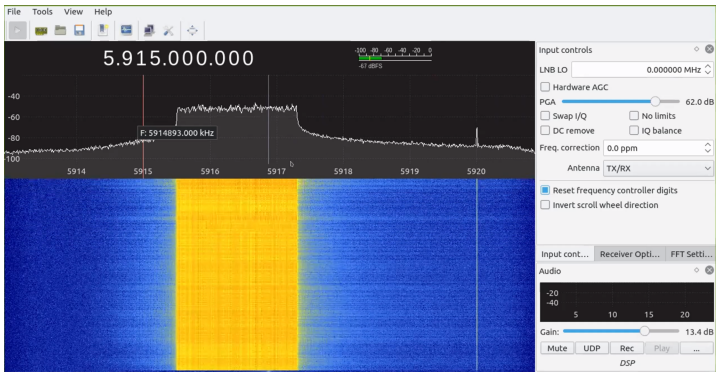


Fig. 3: A screencap of GQRX, displaying C-V2X packet transmission in the 5.915 GHz band

2 Concept Development (Authored By Jason Inirio)

During the semester, we focused on the research aspect of our project. Most of the research done has been through the developments of understanding C-V2X communication protocols and working with software defined radios (SDRs). Recently, we began to use signal analysis, software libraries such as srsRAN, and tools like iperf to better understand and use C-V2X. This allowed us to further our development and create foundations for our next objective; machine learning detection.

To build on a neural network for C-V2X and maintain an adequate level of accuracy - noise must be accounted for. Our motivation for this comes from radios developing large amounts of signals on a 5Ghz band, and thus creating noise. Because there is a massive amount of data every second, our model must be able to also withstand and process any signal correctly to be useful. Since SDRs have an immense range of signals that will cause noise to our model, we will need to utilize a separate algorithm or method to combat and preprocess our noisy data generated from the radios effectively. Eventually, this will require the use of feature mapping and some form of dimensional reduction and analysis. With a successful completion, a machine learning model can be developed to efficiently and accurately detect misbehavior actors in a C-V2X band.

Before developing a machine learning model and accounting for noise in the data, we needed to first realize what type of C-V2X packet layer we would need. With this in mind, we would be able to begin implementing machine learning to detect misbehavior. Conceptually, detecting misbehavior attackers will utilize our signal analysis tools and methods to begin data collection. Next semester, we aim to look more into signal processing using the SDRs, and hopefully learn what packets and information C-V2X sends over the air. To further show improvement of metric in the detection, we may also use software simulations to showcase how our system would behave. We will be planning to use a neural network such as PyTorch, since Python is the most commonly used language for signal analysis and data collection. Automation will also play a key role in how we examine C-V2X communication and data. With automated scripting, we will be able to process and test accurate data before applying it to our machine

learning model. A final goal for the end of next semester is to be able to showcase our machine learning detection system over a web API for ease of use and maximum clarity.

Because the focus of our project is on physical layer anomaly detection, and because we have begun looking at the decoded packets generated from srsRan's `cv2x-traffic-generator`, the natural next step is to see if we can train an ML model on physical layer data. The C-V2X resource selection algorithm is run on each user end at the expiration of the Random Sidelink Resource Re-selection Counter (SLRRC) [7]. Resource selection by normal vehicles rely on the sensing methods of energy level decoding, Sidelink Communication Information (SCI) decoding, and priority information decoding. Because the objective of this project is to detect anomalies on the physical layer, understanding the decoding mechanism of C-V2X is pivotal to the implementation of the three attacker types in [2] and designing our machine learning algorithms. Once we have processed the traffic data, we can featurize the data on (1) signal-to-noise ratios, and (2) the bitfields of the SCI data, and train a neural network to learn the characteristics of normal packet energy levels and normal SCI bitfields.

3 System Description (Authored by Max Ellsworth)

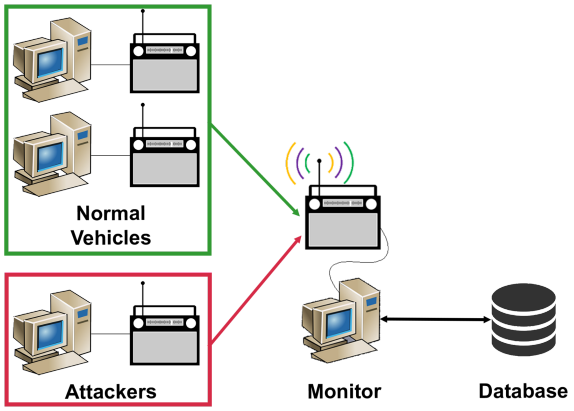


Fig. 4. The hardware setup that will be used to model V2X behavior and attacks. The monitor will listen in on all V2X frequency blocks and save collected data to a web-accessible API. This data can then be visualized and fed into a machine learning algorithm to find attacks.

In the above figure, a few aspects should be considered. First, the monitor. This radio will listen across the entire range of the V2X spectrum, recording activity every few ticks. All compiled information will then be sent to either a PCAP or a database in the final product, which can then easily be parsed by a Python script. The attackers will be attempting to disrupt service of the "normal" vehicles, radios set to transmit simple Basic Safety Messages every few seconds. Other data points, such as the throughput will be taken through occasional executions of the iperf command. As an additional visualization, we will have GNU Radio to help us spot irregularities.

Only three radios are available to our team to model V2X communications. Thankfully, the way in which vehicles divide up the frequency band in V2X allows one radio to simulate multiple vehicles at a time.

4 First Semester Progress (Authored by Max Ellsworth)

This semester, our primary focus was on setting up the infrastructure for the radios, doing background research on the C-V2X protocol, and developing a design plan for our machine learning model. Over the course of two and a half months, the team met almost every Friday at 1 PM with Stefan Gvozdenovic, our graduate student advisor who also serves as a client, at our lab workbench. At these meetings, Stefan would provide us with hands-on instruction using the supporting hardware, i.e., an oscilloscope and two signal generators, necessary for synchronizing the radios and performing tests to verify that the radios were synchronized. Because radio synchronization is a prerequisite for virtually all cellular functionality, about a month was spent on setting up and verifying synchronization. This included troubleshooting issues with creating an over-the-air (OTA) srsRAN cellular client connection to an accompanying base station. While it was unclear what exactly was preventing an OTA connection from happening, a number of combined factors, like synchronization and host performance tuning errors (e.g., CPU cores not being in performance mode), likely contributed to these issues.

Once the cellular base station proof of concept was successful, we spent approximately two weeks performing network connectivity and bitrate tests. The first of these tests was a simple ICMP ping sent from the user end to the base station. This was verified as being successful by having Wireshark open on the base station and pinging the IP address of the base station from the cellular client. Once it was established that the basic network connection was established, a series of iperf tests were run. The purpose of these tests was twofold; first, in addition to demonstrating that the radios could transmit and receive meaningful amounts of traffic over-the-air (i.e., more than a ping), it also allowed us to observe how the frequency spectrum would operate under variable bitrate scenarios. Because iperf can ramp the amount of network traffic up or down depending on what flags are used, it was possible to observe how bitrate (in megabits per second) would affect bandwidth (usually in hundreds of kilohertz to megahertz).

In our first prototype test, our main deliverables were to show proof of an established OTA connection between two USRPs using srsRan and a third USRP that functioned as a monitor. The majority of tests passed with no issue. When executing ping and singly-connected iperf, both the uplink and downlink tests exhibited the expected behaviors, with the UE and ENB issuing "connected" status messages, the USRP transmission and reception LEDs turning on, and both ends recording nonzero bitrates aligned in the direction of traffic. The monitor was able to see communications between the two radios in both packet form via Wireshark and as radio activity on dedicated frequencies via GQRX. The same can be said for multiply-connected iperf in the downlink connection, though the activity on GQRX exceeded the range of the expected frequency band. The only test that deviated from

expectations was multiply-connected iperf in the uplink direction, in which a disconnection during transmission occurred due to the flood of data from the UE. We suspect this is due to a buffer overflow on the base station end. Our setup also changed post-testing in reaction to feedback from Professor Hirsch. The signal generators used to synchronize the UE and ENB previously used a square wave and a pulse with sine modulation. Following the test battery, the square wave was changed to a sine wave, and modulation was removed from both signal generators. After further testing, this has not appeared to have an impact on reliability or clarity of transmission. The disconnection error described above persists, but appears to be relatively rare and is not exclusive to the case of downlink communication. Fortunately, the volume of transmission demanded in multiply-connected iperf exceeds the demands of C-V2X, so this error should not occur in the final design.

5 Technical Plan (Authored by Yixiu Zhu and Samuel Krasnoff)

To complete this project, we must break up our future work into numerous tasks, each with their own milestones. First and foremost is the conclusion of simple C-V2X communication. We currently have one way communication with a V2X signal. To get bidirectional/multicast capabilities, code from both the pssch_ue and ENB libraries will be combined and run synchronously in separate threads to create a full network of V2X capable nodes.

Once a full network is set up, our next challenge is digital signal processing. During our previous experimentations with C-V2X signal collection, We found the file size for a signal that could be detected is enormous. A 6 min signal on the RF spectrum takes more than 500MB of file. For this signal size, it will be really difficult for us to train our machine learning model in a reasonable amount of time. We have experimented with several methods that could possibly help us: change of signal collection method and switching the signal file type. All of those trials make considerably minor space improvements as we expected. We also aim to focus on reducing the noise of our original, raw signal. As our signal collection platform already has an auto weight scaling function, we found these the background noise seemingly came from nowhere. After thoughtful discussion, we decided to implement a feature sampler that could potentially extract features out of our original signal. Currently, we will focus on 5 kinds of features, which are frequency, timing, bit error rate, congestion and signal to noise ratio. With the help of these features, we could potentially further our analysis on the source of the signal noise and also prepare a more reasonable data set for our machine learning training.

While our machine learning models progress and fine tune, we will be targeting to implement DoS attacks simultaneously. We will begin with oblivious attacks, creating another file that utilizes some of the pssch_ue framework and choose random 3 MHz chunks in the 5.9 GHz range, with a PCAP recording all the data that comes through. Following this, we will progress to smart attacks, where the program will periodically scan for signals received from other base stations, and choose those smaller frequency bands to overwhelm. A stretch goal will be to implement coordinated attacks, utilizing each radio to function as multiple base stations. Following the attack implementations, we will transition to the creation of defense mechanisms. Initially, the mitigation will not be algorithmically based, as we familiarize ourselves with changing frequency on the fly through software. If a given SDR fails to receive data for a given number of clock cycles, it will shift its broadcast and receiving frequencies and tell other radios in the network to do the same.

By now, we also have managed to establish a cost-effective defensive strategy when experiencing potential attacks. For our case of targeting DoS attacks, one effective way of mitigation is switching to a different communication channel while attackers are focusing on one specific channel. We make an assumption that an attack is continuous and somewhat periodic and thus will heavily affect our channel switching choices. As we consider using machine learning to solve this problem, we will borrow a simulation environment based on "FLIPIT: The game of stealthy takeover" for our setup: we consider our radio channel as a critical "resource". Ownership of this resource will be changing back and forth between the attacker and defender. As the attacker does the "attack move (DoS attack)", he will possess ownership of this resource. Similarly, when the defender does the "defend move (Switching to a different channel)", the ownership will be switched back to the defender. One special part to note is that this environment is based on a stealthy assumption: attacker and defender will not know the current ownership of the resource until they perform the "attack move" or "defend move". To be more explicit, we can refer to the figure below:

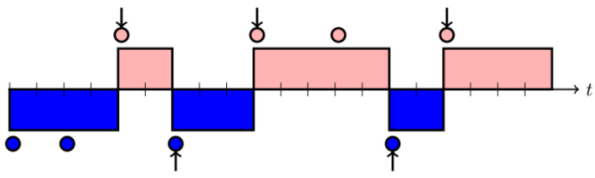


Fig 5: Blue: defender related move & ownership of resource Pink: attacker related move & ownership of resource

With the x-axis as time, we consider blue and pink rectangles as the time of defender's and attacker's ownership of the resources. Blue and red circles represent the time the defender and attacker made their individual move. We also can use the vertical arrow to visualize the time that ownership of resources has switched.

As we finish our environment setup, we strive to develop the most efficient strategy for defenders to own its resources. We will use 2 deep Q-learning networks (DQNs) for our training purpose, more specifically one DQN for the

attacker and another network for the defender. We will functionally give higher rewards based on the time span that one owns the resource. For this case, we started with a non-adaptive strategy and assumed the attacker's attack was periodic. We have several runs based on the simulation-provided data, but for further understanding, we still need our ML detection to be completed. [1]

Categories	Classes of Strategies	
Non-adaptive (NA)	Exponential	Periodic
	Renewal	
	General non-adaptive	
Adaptive (AD)	Last move (LM)	
	Full history (FH)	

Fig. 6: The Hierarchy is ordered (from top to bottom) by increasing the amount of feedback received by a player in the game.

For our future plans, we are more interested in the last move (LM) strategy that attacker and defender learn according to the time when the opponent player played last. We believe these practices will be plenty enough for a transportation based C-V2X DoS defense.

6 Budget Estimate

No.	Project-related Device & Materials	Specs of Device & Material		Total cost
		Item number	Item price ($ per piece)	
1	Desktop with Dragon OS	2	Around 500	1000
2	NI 2901 USRPs (AFRL provided)	3	2010	6030
3	Keysight 33500B signal generators	2	1979	3958
4	LeCroy Wavesurfer 422 oscilloscope	1	5141	5141
			TOTAL	16129

7 Attachments

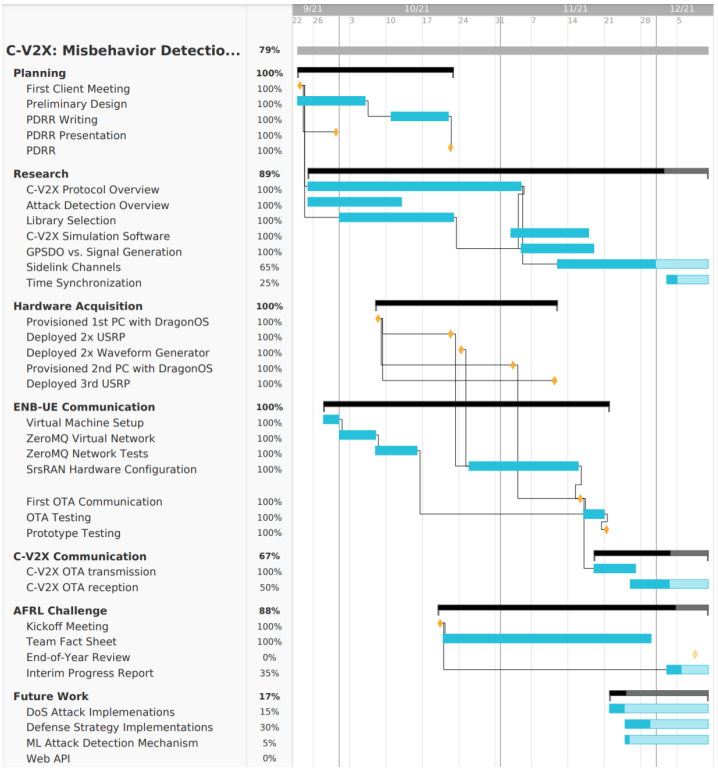
7.1 Appendix 1 – Engineering Requirements (Authored by Julia Zeng)

The model of C-V2X communication must be implemented, attacked, and monitored using only four National Instruments USRP-2901 software-defined radios (SDRs) operating in the 5.9 GHz band. Our goal for our final deliverable is that the SDRs must be capable of regular C-V2X communication with a packet delivery success rate of 60% up to four feet. Due to spatial constraints of the lab setup, we have modified the communication distance from the initial specification of 30 ft; if time permits, we would also like to have the SDRs communicate up to 30 ft, though this is a soft goal. The C-V2X protocol must be simulated faithfully on the SDRs using an existing implementation like the SrsRAN radio suite [3]. There must be a visualization of channel resource usage via a web-accessible API populated by the monitor SDR and accessible 24/7.

There must be three classes of DoS attacks analyzed with samples labelled appropriately to distinguish them. To ensure that our ML algorithm is invariant to any class disproportionality in the dataset, we are aiming for 70% classification accuracy within each DoS attack type – oblivious, smart, and cooperative. These attacks are outlined in DoS Attacks on C-V2X Networks [2]. We aim for 60% accuracy of classification in smart and cooperative attacks for low-vehicle density settings, since this is the most effective attack type in this environment. An AUC value of 0.7 or higher and an F1-score of 0.75 or higher is desirable to favor false positives and minimize false negative detections of malicious attacks.

The misbehavior detection system is to be supported by a batch offline learning model when the live dataset is small and by an online learning model with progression validation when the live dataset becomes larger than the batch model's. Online learning is necessary to account for the temporal aspect of our data.

7.2 Appendix 2 – Gantt Chart (Authored by Michael Aliberti)



[1] Van Dijk, M., Juels, A., Oprea, A. et al. FLIPIT: The Game of "Stealthy Takeover". J Cryptol 26, 655–713 (2013). https://doi.org/10.1007/s00145-012-9134-5

Acknowledgment

The authors wish to thank Professor David Starobinski, Stefan Gvozdenovic, and Tony Lizza (Air Force Research Laboratory) for their continued support of this project.

References

[1] Li, Yang & Hou, Ronghui & Lui, King-Shan & Li, Hui. (2018), "An MEC-Based DoS Attack Detection Mechanism for C-V2X Networks," 1-6. 10.1109/GLOCOM.2018.8647323.

[2] Nataša Trkulja, David Starobinski, and Randall Berry, "Denial-of-Service Attacks on C-V2X Networks," AutoSec 2021, February 2021.

[3] R. Lindstedt, M. Kasparick, J. Pilz and S. Jaeckel, "An Open Software-Defined-Radio Platform for LTE-V2X And Beyond," 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), 2020, pp. 1-5, doi: 10.1109/VTC2020-Fall49728.2020.9348771.

[4] G. Twardokus, and H. Rahbari, "Evaluating V2V Security on an SDR Testbed"

[5] *Wireless Access in Vehicular Environments–Security Services for Appli- cations and Management Messages*, IEEE Standard 1609.2-2016, 2016.

[6] Twardokus, Geoff, "Intelligent Lower-Layer Denial-of-Service Attacks Against Cellular Vehicle-to- Everything" (2021). Thesis. Rochester Institute of Technology.

[7] Shah, Ghayoor & Saifuddin, Md & Fallah, Yaser & Gupta, Somak. (2021). RVE-CV2X: A Scalable Emulation Framework for Real-Time Evaluation of CV2X-Based Connected Vehicle Applications.