A Car Hacking Mini-Unit for High School Cybersecurity Education

Cooper Dean[1], Brian Almaguer[1], Timothy Buck[2], Nichole Kunkle[3], Christian Nguyen[4], Preet Patel[1], Anne Roberts[5], Marti Shirley[6], Huirong Fu[7], Xiaoyan Sun[1], and Jun Dai[1]✉

[1] Worcester Polytechnic Institute
{cjdean, baalmaguer, pnpatel, xsun7, jdai}@wpi.edu
[2] Chesapeake High School
tbuck@aacps.org
[3] Chambersburg Area Senior High School
nichole.kunkle@casdonline.org
[4] Troy High School
cnguyen@fjuhsd.org
[5] Christian Brothers College High School
RobertsA@cbchs.org
[6] Glenbard East High School
marti_shirley@glenbard.org
[7] Oakland University
fu@oakland.edu

Abstract. As technology advances, the need for education on the security of these technologies becomes increasingly critical. However, security education predominantly targets post-secondary curricula, leaving high school learners with limited engaging resources. To address the growing demand for high school cybersecurity education, as highlighted by the College Board's AP Career Kickstart Cybersecurity Pilot Program, this paper bridges the gap by introducing a practical and engaging mini-unit focused on autonomous vehicle security. Autonomous vehicles represent a cutting-edge domain with interconnected systems vulnerable to cyber-attacks, including remote vehicle control, adversarial attacks on vision systems, and exploits on keyless entry systems. Drawing from teaching experience and feedback from five high school educators, we present nine labs designed to teach cybersecurity concepts through hands-on exploration of autonomous vehicle systems. These labs are divided into three sections—CAN bus security, key fob security, and security with AI road image processors—each with progressive levels of complexity. Students engage in activities ranging from simulations of replay attacks and adversarial manipulations on road signs to experimenting with hacking hardware and programming AI models. High-profile real-world incidents, like the 2015 Jeep Cherokee hack and Tesla Model 3 vulnerabilities, highlight the urgent need to integrate these topics into cybersecurity education. Our hands-on labs bring this critical learning experience to high school students, equipping them with the skills and awareness to tackle emerging challenges, and sparking their interest in postsecondary education and career pathways in the field of cybersecurity.

Keywords: Autonomous Vehicles · Security · Education · High School.

1 Introduction

As technological innovation accelerates, the need for education on its security becomes increasingly essential. However, security education has primarily been confined to post-secondary curricula, creating a significant gap at the high school level. In particular, there is a shortage of engaging, hands-on security labs that explore specific subdomains of the technological landscape.

One rapidly evolving technological field is autonomous vehicles, which rely on numerous interconnected systems, each presenting potential vulnerabilities. For instance, a 2015 Jeep Cherokee was found to have a flaw that allowed remote access, enabling hackers to take full control of the vehicle [10]. Similarly, adversarial attacks on road sign recognition systems can mislead autonomous vehicles into misclassifying objects, posing significant safety risks [12]. More recently, Xie et al. discovered that the Tesla Model 3's Key Cards were susceptible to relay attacks [15]. These real-world examples underscore the pressing need to integrate autonomous vehicle security into education, equipping students with both relevant knowledge and awareness, and hands-on learning experiences.

High school cybersecurity education is essential for addressing the growing workforce gap in the field by introducing students to cybersecurity concepts early in their academic journey. While efforts to build the cybersecurity workforce have largely focused on college-level education, expanding these initiatives to high schools can spark students' interest before they choose their majors and career paths. Recognizing this need, the College Board has launched the AP Career Kickstart Cybersecurity Pilot Program [1] for hundreds of high schools to test their cybersecurity frameworks for two new courses: "AP Cyber: Networking" and "AP Cyber: Security". However, the availability of quality and engaging cybersecurity teaching materials remains limited, making it challenging for educators to deliver effective instruction. Many existing resources are either too advanced for high school students or lack the hands-on, interactive components necessary to sustain student interest.

This paper addresses the gap in high school cybersecurity education by introducing a set of nine high school-level labs, newly developed or adapted from existing resources, to teach cybersecurity concepts through the lens of autonomous vehicles. Organized into a structured mini-unit[8], these labs focus on three key components of modern vehicles: the CAN Bus, key fobs, and vision systems. By providing engaging and accessible resources, this initiative aims to equip high school educators with innovative tools to enhance security-related instruction, fostering a deeper understanding of cybersecurity from an early stage.

[8] https://wpi-lions-group.github.io/ADS-Labs/

2 Background and Related Work

2.1 Autonomous Vehicles

CAN Bus. The Controller Area Network (CAN) Bus serves as the central communication hub, transmitting information between various edge devices within a vehicle. Exploiting vulnerabilities in this system can grant attackers access to critical functions such as braking and steering [3,10,13]. To illustrate these security risks, we introduce two hands-on labs focused on replay attacks targeting the CAN Bus. These labs provide students with a practical understanding of real-world cyber threats in automotive systems, enhancing their grasp of critical security vulnerabilities in modern vehicles.

Key Fobs. Modern key fobs must securely communicate with vehicles to prevent theft. However, hackers can exploit vulnerabilities in key fobs using replay attacks [6], relay attacks [15], and rolljam attacks [5]. To explore these threats, we have introduced three labs focused on key fobs. The first two labs introduce replay and rolljam attacks, while the final lab challenges students to design their own secure key fob, protecting it against the attacks they have studied.

Vision Systems. Autonomous vehicles increasingly depend on computer vision systems to see and interpret their surroundings, which must be continually updated to counter emerging threats. Adversarial attacks on road sign images, for example, can interfere with object classification tasks [12]. To address this, we have included three labs focused on adversarial AI attacks targeting road sign classification, enabling students to explore the vulnerabilities and countermeasures within these critical systems.

2.2 Autonomous Vehicles in Cybersecurity Education

In 2019, Payne [13] developed a car hacking lab as part of a semester-long security course. This lab guides students in using open-source tools to perform a replay attack on a virtual CAN Bus. The lab is designed for undergraduate and graduate students, causing the setup and instructions to be too difficult for our use case. Our *CAN Bus Hacking with ICSim* lab is adapted from Payne's *Advanced Car Hacking Lab Assignment*, with streamlined instructions and reduced setup requirements designed to be taught at the advanced high-school level.

Francia et al. [2] examine curriculum challenges and propose a series of hands-on scenarios using real simulation hardware to exploit a CAN bus. To ensure accessibility for high school students, we opted not to adopt or modify these scenarios due to their specific hardware requirements.

McCall et al. [9] constructed an educational activity in which a Raspberry Pi simulates a CAN bus. Students completed three exercises: analyzing dumped CAN data, performing a replay attack similar to Payne's lab [13], and injecting

fake CAN data into the network. Due to the hardware requirements and overlap with Payne's lab, we chose not to adopt or modify these exercises.

Grover et al. [4] explore the integration of AI and Machine Learning in cybersecurity education. After conducting two workshops, they received positive feedback from teachers on incorporating AI into high school classrooms. Similarly, we introduce a set of labs focused on security issues with AI vision systems.

3 Labs

We have introduced a series of nine labs that together form a mini-unit on autonomous vehicle security. These labs are structured into three sections, each focusing on a critical component of an autonomous vehicle: CAN Bus, Key Fobs, and AI Road Image Processors. Each section includes three labs, designed to provide targeted, in-depth exploration of security challenges in that area. We present the labs in our recommended teaching sequence, as outlined in Table 1.

Table 1. The mini-unit plan in recommended sequence

Lab	Section	Version
Hands-On Hack and Defense for an Autonomous Vehicle	Can Bus	Low Tech
CAN Bus Replay Attack Simulation	Can Bus	Med Tech
CAN Bus Hacking with ICSim	Can Bus	High Tech
Real-World Key Fob Cloning	Key Fobs	High Tech
Rolljam Attack Simulation	Key Fobs	Med Tech
Designing a Smart Key Fob Prototype	Key Fobs	Low Tech
Simulating Road Sign Attack with Adversarial Images	AI	Low Tech
Road Sign Recognition with Google Teachable Machine	AI	Med Tech
Real-Time Adversarial Attack on Road Sign Recognition	AI	High Tech

Three of these labs are designed to be completed without the use of a computer or laptop. Instead, they focus on group discussions and creative thinking activities. While some setup with printed materials and art supplies may be needed, these activities emphasize conceptual engagement over technical execution. In Table 1, we designate these labs as "Low Tech" (LT).

Three of these labs are fully online simulations, accessible for free. To support these activities, we have developed two static websites hosted on GitHub Pages, demonstrating different ways to hack a car. Additionally, one lab utilizes Google's Teachable Machine to simulate AI-based road sign recognition. In Table 1, we designate these labs as "Med Tech" (MT).

The final three labs are designed for more advanced students or require specialized equipment, making them a more realistic alternative to the free simulation labs. These labs provide hands-on experience with hardware hacking, working within a virtual machine, and programming an AI model. In Table 1, we designate these labs as "High Tech" (HT).

Please be advised that the three key fob labs should be completed in reverse order. This approach ensures a smoother learning progression, beginning with the High-Tech lab, which is the easiest to grasp conceptually. The Low-Tech lab is introduced later, as it builds on foundational knowledge of key fob security.

3.1 Hands-On Hack and Defense for an Autonomous Vehicle

This lab serves as an introduction to the mini-unit plan and does not necessarily need to cover the CAN Bus. However, it is included in this section since the topic will likely be new to students, serving as an effective starting point. We created the lab with this in mind, using the CAN Bus in some of our examples.

This lab spans two days, beginning with a lesson that defines key terms, introduces autonomous vehicles, and examines a real-life case study. Students will then collaborate in groups to identify potential vulnerabilities in autonomous vehicles and present their findings to the class. For homework, they will research an additional case study to further explore real-world challenges in autonomous vehicle security.

On the second day, students will briefly present their case studies to the class. Following the presentations, they will participate in a role-play simulation, dividing into two groups: the red team and the blue team. The red team will focus on devising methods to penetrate into an autonomous vehicle, while the blue team will strategize defenses against potential attacks. After a brainstorming session, the red team will "attack" by outlining their approach, and the blue team will respond with a defense strategy. This cycle will continue as time allows, concluding with a class discussion on the critical roles both teams play in ensuring vehicle security.

By the end of this lab, students will have a foundational understanding of cybersecurity in the context of autonomous vehicles. They will have explored recent attacks and gained insights into the importance of vehicle security.

3.2 CAN Bus Replay Attack Simulation

This lab was created with inspiration from the "High Tech" lab *CAN Bus Hacking with ICSim*, which will be introduced later. Recognizing the need for a version that requires no additional setup, we developed this lab as a static website[9] hosted on GitHub Pages. With minimal hardware requirements, students only need a computer with internet access to complete the activities.

In this lab, students will conduct a replay attack on a simulated CAN Bus communication network. The simulation allows students to control various components of a car, as shown in Fig. 1. They can send commands over the network to open the hood, a door, and the trunk, as well as start the engine and activate both blinkers. Additionally, students have access to a "Command Log," which enables them to record all commands transmitted over the network.

[9] https://wpi-lions-group.github.io/CAN-Bus-Replay-Attack-Lab/

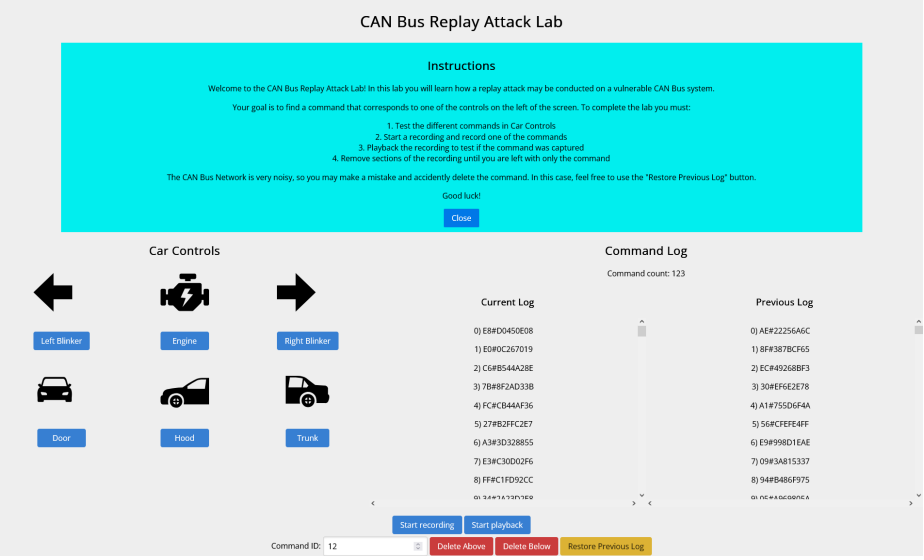

Fig. 1. Screenshot of the "CAN Bus Replay Attack Lab" website. Instructions are shown at the top in blue. Car controls are shown on the left. The command log is shown on the right. The lab control buttons are shown at the bottom of the page.

Using the command log and a set of control buttons, students can replay recorded commands and remove specific entries from the log. The objective of the lab is to narrow down the list of commands to successfully execute a replay attack on one specific car control. For instance, a student aiming to open the car door would first start a recording and trigger the door-opening command. Upon playback, they might observe that the door opens after 80 commands have been sent. To isolate the target command, the student will incrementally delete the first 80 commands using the input form. This process of playback and command elimination continues until only a handful of commands remain. The student may then test each command individually until the target command is found. If an error occurs and the target command is accidentally removed, students can restore the previous log and attempt the process again.

By the end of this lab, students will see a simulated CAN Bus in action and successfully execute a basic replay attack. They will develop an understanding of the CAN Bus's purpose, functionality, and its appeal as a target for hackers. A post-lab discussion should explore the potential risks associated with a compromised CAN Bus and examine security measures that can be implemented to safeguard vehicle systems.

Fig. 2. Screenshot of a Kali Linux virtual machine running ICSim. The left two windows show the ICSim simulator (top) and controller (bottom). The right window shows the linux terminal.

3.3 CAN Bus Hacking with ICSim

Our adaptation of Payne's car hacking lab [13] was based on a 2022 blog post from OffSec [14]. The lab runs on Kali Linux[10] using VirtualBox[11]. However, we found that this version was outdated requiring additional setup beyond the original instructions. To simplify the process, we introduce an ISO file[12] that functions as an external hard drive for the virtual machine. This ISO contains a folder with a snapshot of most required dependencies, allowing students to complete the setup by simply attaching the ISO to the virtual machine, copying the necessary files, and installing a single additional package. We also included our own set of instructions.

Building on the previous lab, students will once again perform a replay attack on a simulated CAN Bus. However, instead of using buttons and a visual command list, they will work directly in the Linux terminal to capture and replay packets containing commands, as illustrated in Fig. 2. First, students will be prompted to record the noisy CAN Bus network and examine the length of the log file. Within just a few seconds, thousands of simulated commands may be transmitted over the network. Next, they will be asked to use a controller to execute specific actions on the car. For example, a student can press combina-

[10] https://www.kali.org/
[11] https://www.virtualbox.org/
[12] https://github.com/WPI-LIONS-Group/ADS-Labs/releases/

tions of Left Shift, Right Shift, and the A, B, X, and Y keys to open and close the car's doors. After experimenting with multiple controls, students will select a specific target command for their attack. They will then capture and isolate that command in a recording to carry out the replay attack.

After recording a log file, students can begin isolating the target command by systematically removing commands from the file. To aid in this process, they are given access to various Linux terminal commands, including *wc*, *head*, *tail*, *cat*, and *echo*. Additionally, they can replay any log file to verify whether the target command was successfully captured. A common strategy at this stage is to iteratively remove half of the file and test whether the target command is present in either the first or second half.

By completing this lab, students will gain an understanding of the CAN Bus and its potential vulnerability to replay attacks. They will also develop hands-on experience with the Linux terminal and commands within a virtual machine.

3.4 Real-World Key Fob Cloning

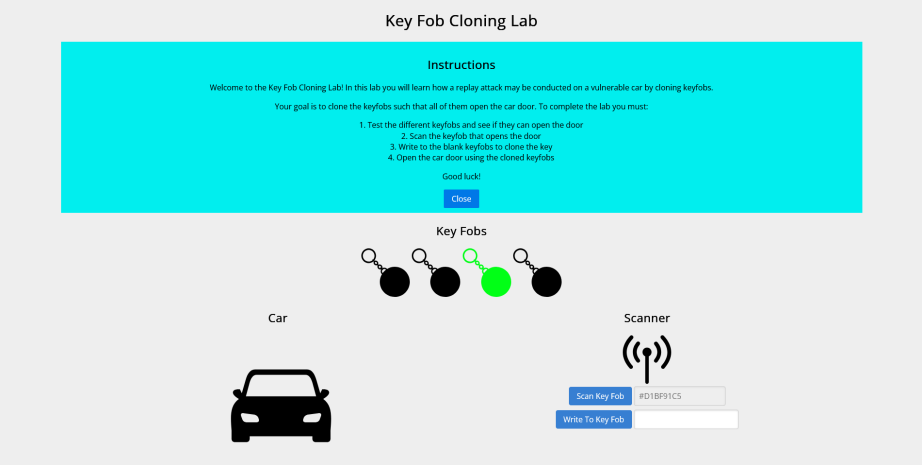

Fig. 3. Screenshot of the "Key Fob Cloning Lab" website. Instructions are shown at the top in blue. Four key fob icons are in the center. A simulated car door is shown on the left. The RFID scanner with read and write operations is shown on the right.

The most basic attack on an RFID key fob is to clone its signal onto a blank fob [7]. This process requires minimal hardware and, in its simplest form, can be accomplished using a single RFID scanner. For this lab, we selected the Flipper Zero[13] due to its ease of use and educational value. When available, this device

[13] https://flipperzero.one/

can serve as a valuable teaching tool, enabling educators to develop a variety of lessons around its applications.

In this lab, students will use a Flipper Zero device to clone a vulnerable key fob. The lab involves a few quick steps. First, the instructor pre-programs a blank key fob with an RFID signal. Students then use a Flipper Zero or an alternative RFID reader to capture the signal. Once saved, they will clone the signal onto a new key fob, successfully replicating the original access credential.

Conducting this lab with real equipment in a classroom setting may be cost-prohibitive. Hence, a free and web-based version has been developed by us and hosted on GitHub[14]. As shown in Fig. 3, this simulation includes multiple key fobs, a car, and a scanner. One key fob contains the RFID signal required to unlock the car door, while the other three are blank. Students can select a key fob, use the scanner to read or write a signal, and then test it by attempting to unlock the car door.

By completing this lab, students will show a basic understanding of RFID signals through the context of key fobs. They will understand why key fobs require basic security features to avoid simple cloning attacks.

3.5 Rolljam Attack Simulation

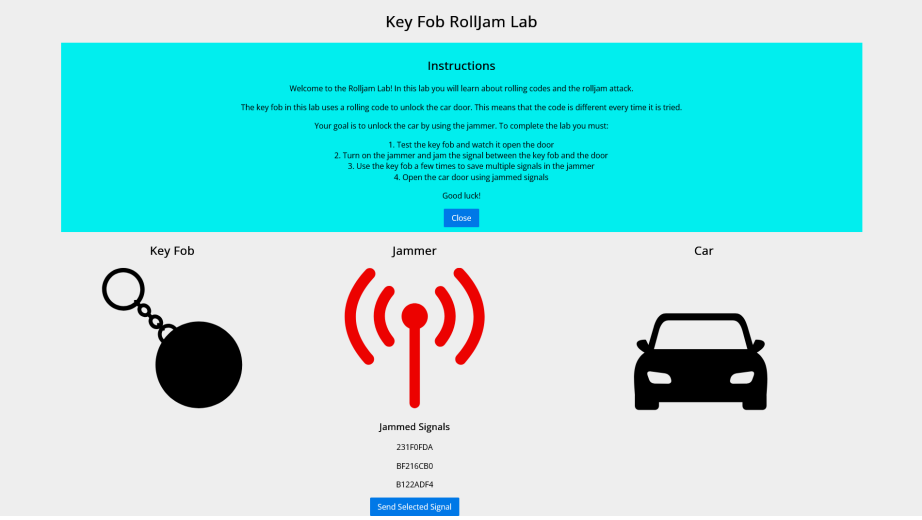

Fig. 4. Screenshot of the "Key Fob Rolljam Lab" website. Instructions are shown at the top in blue. A clickable key fob is shown on the left. The signal jammer is shown in the center with stored signals below. The simulated car is shown on the right.

[14] https://wpi-lions-group.github.io/Key-Fob-Cloning-Lab/

The rolljam attack, first demonstrated by Samy Kamkar at DEFCON 2015 [5], exploits a jammer to bypass rolling code security in key fobs. When a key fob sends a signal, it generates a unique code that changes (or "rolls") with each use. This dynamic code is synchronized with the receiver (such as a car or door lock). Even if an attacker intercepts a signal, the code will no longer be valid the next time it is transmitted because it will have already advanced to a new value.

To provide a learning experience with minimal cost of setup and maintenance, we developed the Rolljam Attack Simulation lab as a static website[15]. In this lab, students will seek to bypass a key fob's rolling code system using a signal jammer. As illustrated in Fig. 4, the simulation includes three interactive components: a key fob, a jammer, and a car. Selecting the key fob attempts to send an unlock signal to the car. The jammer, which can be toggled on or off with a visual indicator, intercepts these signals—either blocking them or forwarding them to the car. All blocked signals are recorded in a "Jammed Signals" list, allowing students to analyze and reuse them to bypass the rolling code security.

The objective of this lab is to unlock a car door without using the key fob. First, students will verify that the key fob successfully opens the car door when the jammer is inactive. Next, they will activate the jammer to block the key fob's signal, capturing a series of "Jammed Signals" recorded as hexadecimal strings. Students will then select a captured signal and transmit it to the car. Since the key fob operates with a rolling code system, only the correct sequential signal will successfully unlock the door.

Students who complete this lab will gain an understanding of rolling codes in key fobs and their potential vulnerabilities. After the lab, a discussion should be conducted on possible security enhancements to mitigate these risks.

3.6 Designing a Smart Key Fob Prototype

Following the completion of the previous two labs, students should have a solid understanding of common attacks on key fobs. This lab, designed for individual activity or group work, encourages creative thinking in designing a key fob prototype. Using their knowledge of attacks and security measures, students are tasked to develop a key fob that is resistant to the vulnerabilities demonstrated in earlier labs.

To complete this lab, students will develop a conceptual prototype of a key fob, presented as a detailed description rather than a physical product. This prototype is for demonstration purposes only and does not require any electronic parts. Alongside their description, students should provide design notes outlining the specific security features they have incorporated. Finally, a discussion should be conducted to assess potential vulnerabilities in the proposed key fob designs.

Students should have the opportunity to refine their designs based on the discussion. Later, they will present their prototypes to the class, highlighting their security features, initial vulnerabilities, and the modifications made after

[15] https://wpi-lions-group.github.io/Key-Fob-Rolljam-Lab/

the discussion. Following the presentations, students will engage in a broader discussion on key fob technology and the importance of security in its design.

3.7 Simulating Road Sign Attack with Adversarial Images

This is a lab activity that does not requires computer access. Students will use stickers, markers, and other materials to make subtle modifications to a printed road sign. They will then be asked to discuss how these slight alterations may affect a car's vision model, and how this can be avoided.

This lab is divided into three parts: an introduction, a simulation, and a discussion. The introduction allows a teacher to explain AI image processing in autonomous vehicles, focusing on how these systems interpret road signs. The teacher also introduces adversarial attacks, emphasizing how small modifications to a road sign can significantly impact the functionality of a car's vision model.

The second part of the lab is a simulation where students engage in road sign identification and adversarial modifications. Working in small groups or individually, they receive a printed road sign and discuss or document how an autonomous vehicle might recognize and interpret it based on its design, symbols, and text. After this initial analysis, students use stickers and markers to apply subtle adversarial changes to the sign, ensuring that while the modifications do not obscure its meaning to a human observer, they have the potential to mislead an AI model. Through this part, students explore the vulnerabilities of machine vision and gain a deeper understanding of how seemingly minor alterations can disrupt an AI system's ability to process visual information accurately.

The last part of the lab is the discussion and analysis. Each group presents their modified road sign, detailing the changes they made and how these alterations could impact an AI system's interpretation. They also discuss the potential risks associated with an autonomous vehicle misidentifying a road sign due to these modifications. The discussion will also inspire students to explore defense mechanisms against adversarial attacks, prompting students to consider potential solutions for enhancing AI robustness and reliability.

By completing this lab, students will develop a foundational understanding of image processing and adversarial attacks in the context of image-based object identification and classfication. They will also become acquainted with the risks of misidentifying road signs and explore strategies for mitigating such attacks.

3.8 Road Sign Recognition with Teachable Machine

Building on the previous lab, this exercise takes students a step further by enabling them to train their own road sign image recognition model using Google's Teachable Machine[16]. After following the provided instructions to train the model, students are encouraged to creatively test its robustness by introducing adversarial changes to the road sign images. As illustrated in Fig. 5, writing

[16] https://teachablemachine.withgoogle.com/

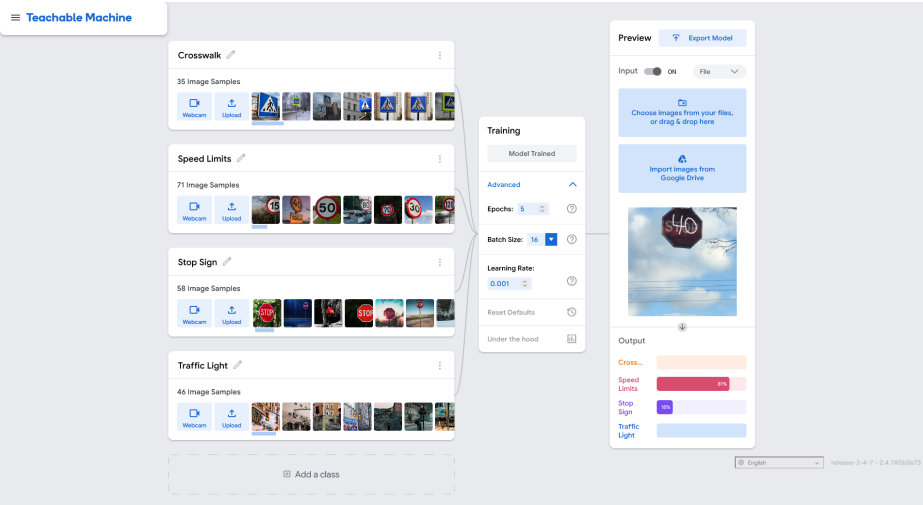

Fig. 5. Screenshot of Teachable Machine. The input data is shown on the left split into four classes. Training settings are shown in the center. The model preview is shown on the right incorrectly classifying a stop sign with adversarial markings as a speed limit sign.

a number on a poorly placed stop sign can deceive the model into misclassifying it as a speed limit sign.

For this lab, we provide a dataset of 230 images, which is divided into 210 images for training the model and 20 images for testing. The dataset consists of four classes: *crosswalks*, *speed limits*, *stop signs*, and *traffic lights*. The class distribution is detailed in Table 2. This dataset was curated by manually selecting images from a larger public domain Road Sign Detection dataset [8], ensuring they are suitable for this classification task. The dataset is hosted on GitHub for easy access[17], available as a ZIP file with separate folders for each class, allowing students to access the images in batch.

Table 2. Road Sign Image Dataset Classes

Class	Training Count	Test Count
Crosswalk	35	5
Speed Limit	71	5
Stop Sign	58	5
Traffic Light	46	5

[17] https://github.com/WPI-LIONS-Group/Road-Sign-Image-Dataset

By completing this lab, students will understand and be able to train and test their own computer vision models. They will also explore creative methods to deceive their models by applying subtle adversarial modifications.

3.9 Real-Time Adversarial Attack on Road Sign Recognition

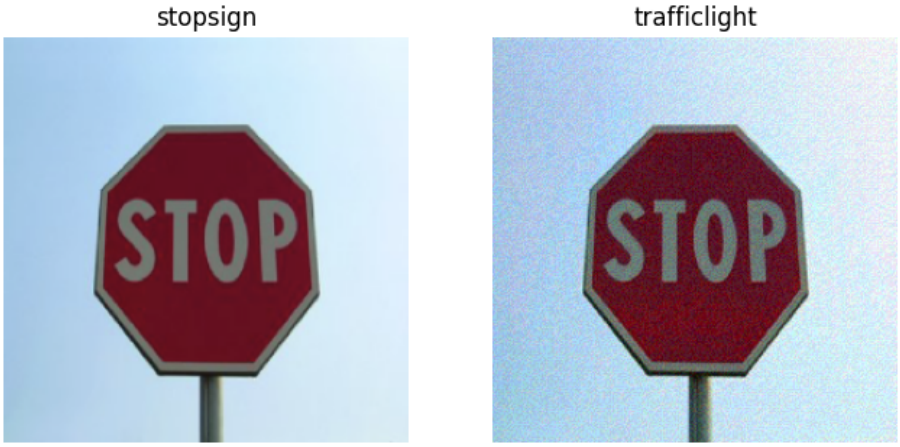

Fig. 6. Left: AI model correctly identifying a stop sign. Right: AI model misclassifying a stop sign after using applying the fast gradient method with CleverHans.

This lab allows students to experience real-time adversarial attacks on road sign recognition. A similar attack can be executed by automatically generating adversarial examples using the CleverHans library [11]. We designed this lab as a tutorial that integrates CleverHans with a Google Colab[18] notebook. Using the dataset from the previous lab, students are guided step by step to build an AI model that recognizes the images and then automatically generates adversarial examples to challenge the model.

The tutorial is designed to guide students through the process of completing the lab. We assume high school students may not be familiar with the tools used, and hence all the code is provided directly in a markdown file, making it easy to copy and paste into a Colab notebook.

The lab is divided into three parts. First, students will install and import the required packages before loading the images from our dataset for use in the model. Next, they will create an AI model using TensorFlow[19], train the model, and observe its performance on the test data. After initial training, students will use the CleverHans library to generate adversarial examples for the model and

[18] https://colab.research.google.com/
[19] https://www.tensorflow.org/

analyze the results. Finally, students will have the option to adjust the model's parameters and retrain it in an attempt to improve its security.

Upon completing the lab, students will understand how images are interpreted by a computer, how an AI model can be trained to classify these images, and how small alterations to the images can generate adversarial examples that deceive the model. Following the lab, a discussion should focus on potential security measures for AI models and the importance of adversarial testing.

4 Evaluation Plan

This paper primarily focuses on the content introduction of the lab set. In addition, we have developed a robust evaluation plan for piloting these labs in high school environments. The project team includes five high school teachers who have been involved from the initial design phase and will play an active role in the evaluation process. The evaluation plan will include the following:

Teacher Feedback: The five participating teachers will provide feedback on the lab's clarity, engagement, and educational value. They will assess how effectively the lab content aligns with the curriculum and its ability to facilitate learning.

Student Performance: The lab's impact on student understanding will be evaluated through assessments of students' knowledge before and after completing the lab. Teachers will track students' ability to grasp concepts related to vehicle security, focusing on attacks and countermeasures.

Lab Usability: Teachers will evaluate the ease of use of the lab materials, including the clarity of instructions, the accessibility of resources, and the overall technical setup. Any challenges or obstacles encountered during the lab usage will be documented.

Student Engagement: Teachers will monitor student engagement during the lab exercises, observing how students interact with the tools, peers, and apply their learning. Feedback will be gathered through surveys and informal discussions with students.

Suggestions for Improvement: Teachers will collect suggestions from students regarding the lab's content, tools, and overall structure. This feedback will be used to refine the lab for future refinement and use.

After piloting in labs in high school classrooms, discussions with the teachers will be held to evaluate the overall success of the lab and explore areas for improvement. By integrating the teachers into the evaluation process, the plan ensures that the lab set is refined based on real-world classroom feedback and improves its educational effectiveness.

5 Conclusion

In conclusion, as technology continues to evolve, the importance of early cybersecurity education becomes ever more pressing. This paper presents a solution to the gap in high school cybersecurity resources by offering a hands-on, engaging

mini-unit on autonomous vehicle security. Through practical labs that simulate real-world attacks and vulnerabilities, students gain valuable skills and insights into emerging threats. The collaboration with high school educators ensures that the labs are both relevant and effective, providing a foundation for students to explore cybersecurity careers and further studies. By fostering early awareness and critical thinking in this vital field, we empower the next generation to confront the cybersecurity challenges of an increasingly interconnected world.

Acknowledgments. Drs. Xiaoyan Sun and Jun Dai are supported by NSF DGE-2409851. Dr. Jun Dai is also supported by NSF DGE-1934285/2403603.

References

1. Board, C.: Ap career kickstart cybersecurity pilots. https://apcentral.collegeboard.org/about-ap/outreach-and-collaborations/career-kickstart/cybersecurity-pilots (2023), [Online; accessed 1-Feb-2025]
2. Francia, G., El-Sheikh, E., Chi, H.: Vehicle security learning tools and scenarios. In: 2020 International Conference on Computational Science and Computational Intelligence (CSCI). pp. 88–92 (2020). https://doi.org/10.1109/CSCI51800.2020.00022
3. Giannaros, A., Karras, A., Theodorakopoulos, L., Karras, C., Kranias, P., Schizas, N., Kalogeratos, G., Tsolis, D.: Autonomous vehicles: Sophisticated attacks, safety issues, challenges, open topics, blockchain, and future directions. Journal of Cybersecurity and Privacy **3**(3), 493–543 (2023). https://doi.org/10.3390/jcp3030025, https://www.mdpi.com/2624-800X/3/3/25
4. Grover, S., Broll, B., Babb, D.: Cybersecurity education in the age of ai: Integrating ai learning into cybersecurity high school curricula. p. 980–986. SIGCSE 2023, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3545945.3569750
5. Kamkar, S.: Drive it like you hacked it: New attacks and tools to wirelessly steal cars. Presentation at DEFCON **23**, 10 (2015)
6. Kapulica, A., Dakić, V., Morić, Z., Petrunić, R.: Key fob replay attacks on personal vehicles: Vulnerabilities and mitigation strategies. In: 2024 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). pp. 1–5 (2024). https://doi.org/10.1109/HORA61326.2024.10550523
7. Laurie, A.: Practical attacks against rfid. Network Security **2007**(9), 4–7 (2007)
8. Maranhão, A.: Road signs dataset. https://www.kaggle.com/datasets/andrewmvd\/road-sign-detection
9. McCall, S., Yucel, C., Katos, V.: Education in cyber physical systems security: The case of connected autonomous vehicles. In: 2021 IEEE Global Engineering Education Conference (EDUCON). pp. 1379–1385 (2021). https://doi.org/10.1109/EDUCON46332.2021.9454086
10. Miller, C.: Remote exploitation of an unaltered passenger vehicle. Black Hat (2015)
11. Nicolas Papernot, e.a.: Technical report on the cleverhans v2.1.0 adversarial examples library. arXiv preprint arXiv:1610.00768 (2018)
12. Pavlitska, S., Lambing, N., Zöllner, J.M.: Adversarial attacks on traffic sign recognition: A survey. In: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME). pp. 1–6 (2023). https://doi.org/10.1109/ICECCME57830.2023.10252727

13. Payne, B.R.: Car hacking: Accessing and exploiting the can bus protocol. Journal of Cybersecurity Education, Research and Practice **2019**(1), 5 (2019)
14. Team, O.: Introduction to car hacking: The can bus. `https://www.offsec.com/blog/introduction-to-car-hacking-the-can-bus/` (2022), [Online; accessed 24-January-2025]
15. Xie, Xinyi, e.a.: Access your tesla without your awareness: Compromising keyless entry system of model 3. In: NDSS (2023)