PASTA: Portable Automotive Security Testbed with Adaptability

Abstract

For accelerating the development of sophisticated driving-assist technologies such as automated driving, securing vehicles against cyberattacks is challenging. To promote the development of security-measurement methods, a company's electronic control unit (ECU) places tough restrictions on security analysis. In such circumstances, we need an environment that includes transparent ECUs with high adaptability. Ideally, anyone will be able to apply technology and evaluate that technology with ECUs. Simulating an actual vehicle through hardware is also required for assessing threats of cyberattacks. We need not only to provide an adaptable platform for developing measures for existing cybersecurity but also simulate any function in actual vehicles using white-box ECUs. In addition, to easily demonstrate and evaluate the applied security technique, it is important to make the environment portable. Considering these requirements, we propose the portable automotive security testbed with adaptability (PASTA), and give an example for evaluating it for proof of concept. PASTA has the possibility to contribute to a comprehensive development platform against vehicle cyberattacks.

1. Introduction

1.1 Background of this study

Information and communication technology (ICT) in vehicles has been progressing based on expectations for not only providing information to drivers but also for developing self-driving vehicles. Connected technology or full self-driving automatic operation technology is reaching the stage of practical application. However, diversification of services using ICT and diversification of cybersecurity threats are correlated and directly linked to higher risk to drivers. Therefore, the importance of evaluating cybersecurity solutions for vehicles is increasing.

An electronic control unit (ECU) is at the center of the development of cybersecurity technology, and each supplier develops its own automotive information security technology for ECUs in its own environment. To the best of our knowledge, software and design of ECUs in commercial vehicles are commonly concealed to protect intellectual property (black-box ECUs). Because of this, when we evaluate the security of ECUs and commercial vehicles equipped with such ECUs, we have to have a contract with automakers or suppliers of ECUs in most cases, furthermore, if sufficient information on ECUs is not provided by them, time-consuming analysis is required. Attempts to make a secure platform by making it into a black-box platform are common, but this makes it difficult to accurately evaluate the security technology implemented there.

Generally, black-box ECUs have three major drawbacks. First, because the internal design is not disclosed, it is difficult to analyze how the ECUs are programmed. Second, it is difficult for external researchers and engineers to evaluate the effectiveness and safety of the security technology applied to black-box ECUs. Third, since black-box ECUs are not programmable, it is impossible to reconnect ECUs or apply improved patches.

Vehicles incorporating ECUs or ECUs themselves need to protect intellectual property, so there is no need to clarify everything. However, improvement in security technology requires the participation of many researchers and engineers. Therefore, an open and customizable platform for evaluating the security of security technology is required. A feasible solution to this problem is a vehicle simulator consisting of ECU modules with which anyone can develop and analyze their internal communications (white-box ECUs). A white-box ECU is flexible, which allows us to simulate various vehicle communications because of its programmability. In addition, it is expected to have attack surfaces such as JTAG and RS232C. This feature allows us to evaluate cybersecurity attacks assuming physical contacts including disassembling and analysis of power consumption and electromagnetic waves.

In implementing a vehicle model, the most important roles of the hardware in the vehicle is related to "running, turning, and stopping". Reproducing the driving experience regarding such hardware is also an important factor for a testbed. A virtual car that can reproduce driving experience by connecting a highly flexible white-box ECU and a hardware simulator is also necessary. Human-resource development related to cybersecurity is also a problem.

We introduce our portable testbed for automotive security with adaptability and portability (PASTA) as a possible solution to these problems.

2. Previous Work and issues

In this section, we introduce previous work on automotive security testbeds and discuss those issues.

2.1 Previous Work

Previous work 1 — immovable and unaffordable high-quality simulator

In 2013, the HRL Laboratories and GM announced that they were developing an automotive testbed equipped with ECUs, radars, an automobile simulator, etc. [1]. The testbed is equipped with a high-quality simulation environment for actual cars and uses ECUs that are identical to the testbed used in their products. As a result, the automobile simulator provides a high degree of accuracy in simulating vehicles. However, this testbed is immovable and very expensive. Moreover, it is not applicable to other types of vehicles.

Previous work 2 — affordable but unreproducible real simulator

In 2014, Miller and Valasek customized an off-road go-cart and developed a test platform that operates to emulate actual vehicles [2]. They claim that this modified go-cart provides a more affordable experimental environment than using commercial vehicles. Since this experimental environment is implemented at low cost, there may be little financial risk in conducting reproduction experiments.

Contrary to the simulator developed by HRL laboratories and GM, this experimental environment is inexpensive and portable. However, it is generally difficult to remodel a go-cart. Furthermore, since it is an actual vehicle, there is risk of accidents when conducting security experiments.

Previous work 3 — complete software simulation without actual vehicle

In 2011, Munera *et al*. announced a software vehicle simulator for their testbed[3]. They simulated vehicular ad hoc networks (VANETs), also known as vehicle-to-vehicle (V2V) communication. Their simulator uses an extensible open-source tool (called VanSimFM), which is an extension of the current network simulator NS-2 using open source software such as SUMO and CityMob.

Their testbed is highly regarded in that it provides a framework for simulating the V2V network and its evaluation. However, it seems that there are many immature components of their testbed for vehicle cybersecurity evaluation. Their report was limited to the provision of platforms, and we could not confirm that actual attacks can be evaluated or countermeasures can be proposed. Their simulator emulates ECUs internally and simulates hardware behavior such as that of wheels and engines on software. The simulator is inexpensive and is easy to reproduce experiments. However, since the simulator is completely made of software, physical inputs or outputs cannot be evaluated.

2.2 Underlying problems and possible solutions

As we mentioned above, there is no testbed that is affordable and highly adaptable for researchers. Current testbeds that simulate vehicles with high precision are not versatile and are expensive, and a testbed developed inexpensively is difficult to reproduce. Furthermore, none of the testbeds announced thus far use white-box ECUs.

Flexibility is an important factor in a simulator. ECU flexibility means that the ECU can be programmed and the in-vehicle network can be freely designed. Of course, the ECUs used differ depending on the type of vehicle, and vehicle behavior will change if the combination differs in the same ECU. A vehicle simulator needs to be flexible so that it can cope with such differences.

It is not impossible to use black-box ECUs for cybersecurity research. For example, if a research group and original equipment manufacturer (OEM) or supplier collaborate on research, the research group can conduct research using a black-box ECU. However, without cooperation from suppliers (often different from OEMs) in developing black-box ECUs, the scope of investigation is limited. The Automotive Information Sharing and Analysis Center (AUTO-ISAC) provides a system that shares security incidents [4], but there are problems of intellectual property and interaction of ECUs. Therefore, it will not be easy to close the security holes of the testbed based on information obtained from AUTO-ISAC. Because of this, it is difficult to comprehensively address security holes using black-box ECUs. Even if the security-hole problem is solved in one system, similar security holes may remain in other systems.

White-box ECUs overcome the drawbacks of black-box ECUs and enable the following three actions. First, we can evaluate the security of ECUs in various ways such as observing input and output of ECUs or disassemble the program from ECUs without any contract with OEMs or suppliers. Second, because they are programmable, we can apply newly developed security technology in the assumed in-vehicle network, arrange ECUs freely, and evaluate the security technology against cyberattacks. Third, by freely changing the controller area network (CAN) ID, payload, or transmission cycle in the in-vehicle network, it becomes possible to reproduce a commercial vehicle or the functional changes of the ECU through estimating the commercial vehicle's functions and programming ECUs.

ECUs are major targets of cyberattacks. A typical example is a side-channel attack. With this attack, for instance, secret keys in the memory of ECUs are typically extracted from analysis of power consumption or leaked electromagnetic waves. Although the function of ECUs can be reproduced with the software simulator regarding side-channel attacks, a hardware ECU is required for simulation, and countermeasures for such attacks need to be prepared.

Miller et al. developed their testbed by modifying a commercial vehicle. However, this vehicle is constructed by combining black-box ECUs, and it is large because there is an actual actuator. Therefore, there is a high risk to researchers because there is a possibility of unexpected behavior of the actuator when reproducing cyberattacks. To avoid such unexpected accidents, installing the actuator on the testbed should be avoided and a software vehicle simulator or scale model of a vehicle should be adopted to confirm the behavior of the vehicle. Furthermore, since portability is improved by reducing the size of the testbed, it is possible to study and demonstrate even in a small space or indoors.

2.3 Requirements for better Testbed

From the above consideration, it is necessary to satisfy a few requirements to develop a better testbed. We summarize these requirements below.

・White-box ECU:

With a white-box ECU, we can program on actual hardware. Additionally, the in-vehicle network can be flexibly designed similar to that of an actual vehicle. Therefore, it is possible to apply and evaluate security technology in an almost realistic environment. Also, it is based on open technology; therefore, we can evaluate security without any contract with stakeholders such as OEMs and suppliers.

・Simulation of Vehicle:

By avoiding adopting full-size actuators in the testbed and adopting a software vehicle simulator or scale model of a vehicle, even if the vehicle behaves in unexpectedly when being attacked, it is possible to study in safety. Also, since it is expected that the vehicle will be smaller, portability will be improved.

3. Development of Security Testbed

From the discussion in Section 2, we clarified a few necessary requirements that the testbed should satisfy regarding vehicle-security-technology development and useful educational tools. In this section, we present our testbed, i.e., PASTA that meets these requirements.

3.1 Modelling of vehicles

Simulators such as hardware-in-the-loop (HIL) and software-in-the-loop (SIL), which has been used for developing ECUs, are able to cope with various assumed driving scenarios. However, a testbed required for research and development in security as well as learning new security technologies should be more affordable and programmable.

To meet these requirements, we eliminated components not required for a testbed such as expensive sensors and simulators related to vehicle characteristics, e.g., speed, angle of tires, shift position, and status of headlight and brake lights. For automotive security research, expensive and sophisticated simulators and sensors are not necessarily required. Therefore, by omitting these by default, we reduce cost and make it easy to develop a testbed. However, when a testbed is too simple, it is impossible to conduct investigations required with an advanced simulator or on attacks on a specific sensor. Therefore, by preparing interfaces with external devices, we make these additional investigations possible.

When developing a testbed, remodeling an actual vehicle by rewriting ECUs is a straightforward approach. However, when using an actual vehicle, the generalizability of the model may depend on the original vehicle. Therefore, we did not adopt this method but developed our own ECUs so as not to depend on actual vehicle specifications.

Safety is important for experiments on vehicle actuators. Since we cannot predict how the actuator will react to cyberattacks, we need to consider the risk of accidents. To avoid such risk, the actuators deployed in the testbed are not actual size but emulated using software or downscaled.

Another reason for developing such a testbed is to educate cybersecurity professionals. Compared to the increasing number of cyberattacks on vehicles, there are relatively few human resources on the protection side. Due to vehicles having many physically vulnerable points, cyber hackings occur in various ways including security holes due to software bugs and inappropriate design from human error. Education using a physical platform is essential to cybersecurity professionals, and one of the main objectives of developing testbeds is to meet this requirement.

There are currently no model-vehicle testbeds for meeting the above requirements. Therefore, we developed PASTA, which consists of easily accessible components, is adaptable, and can reproduce the required physical environment.

3.2 Overview of PASTA

PASTA has two features that enable practical research, development, and evaluation of vehicle cybersecurity technologies.

First, we used non-proprietary technologies to develop PASTA. Researchers can customize it themselves. Second, some of the internal operations of PASTA are displayed in three formats: 1) status display on the attached monitor, 2) physically moving a scale-model vehicle, and 3) with the software vehicle simulator.

Figure 1 shows an overview of PASTA. Inputs

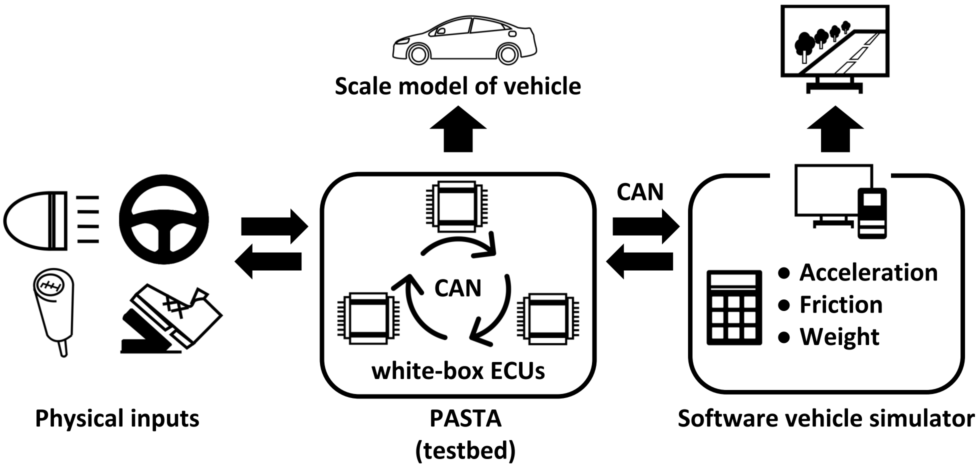

Figure 1. Overview of system around PASTA

from physical equipment are transferred to PASTA, and white-box ECUs are connected to each other via CAN. PASTA is connected to the software vehicle simulator. In the simulator, a computer calculates the characteristics of actual vehicles such as acceleration, friction, and weight, and the simulator displays the behavior of the vehicle by received CAN messages. A scale model of a vehicle connects to the testbed, and the attack results can be displayed from the viewpoint of the vehicle.

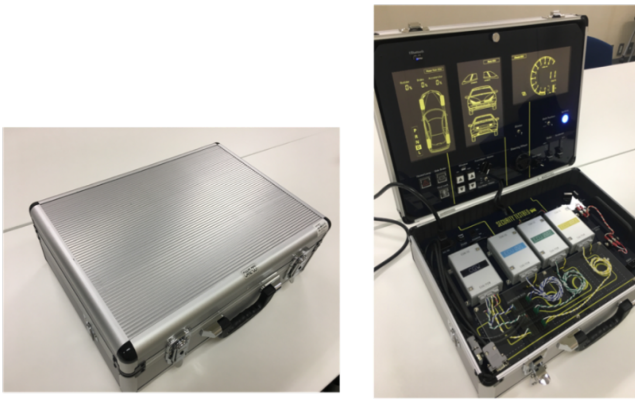

Figure 2: PASTA in attaché case

3.2.1 Feature 1: Using non-proprietary technologies

To improve the adaptability of PASTA, we developed white-box ECUs that can open specifications by using general-purpose components. Users can easily apply their proposed security technology and design the in-vehicle network flexibly. Details of our white-box ECUs and security technology are described below.

White-Box ECU

There are several companies that sell programmable ECUs, but in many cases, they are very expensive. In addition, the programming language may be specially designed or uncommon. We originally designed white-box ECUs using general-purpose components, so they can be affordably constructed. In addition, we used C as the development language, making it possible to rewrite the default program.

Figure 3 shows one of our ECUs designed and developed for PASTA. The IC chips on board are not for automotive use but for general purpose. A microcomputer by Renesas is used for the base. Unlike ECUs using automotive microcomputers, when developing software for ECUs equipped with this general-purpose microcomputer, there is no need to make non-disclosure contracts with microcomputer vendors.

Because our ECUs use a general-purpose microcomputer, it is easy to prepare the development environment. Since these ECUs can be programmed in C, it is possible for users to implement their proposed security technology in a short time. Since PASTA is equipped with attack surfaces, it is also easy to evaluate its security technology. Even though they use a general-purpose microcomputer, if users discover a vulnerability of the microcomputer, they have to report it to the correct facility in the appropriate manner.

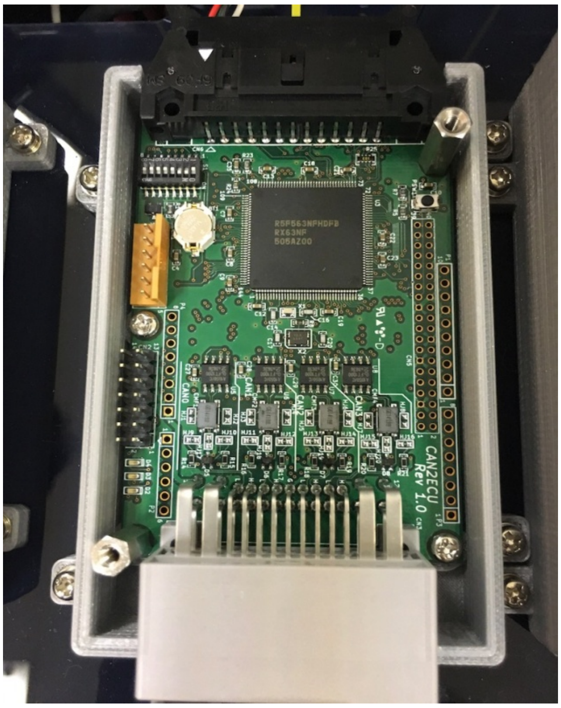

Figure 3. White-box ECU for PASTA

Connection of ECUs in PASTA

In the bottom half of the attaché case, there are four ECUs. In the default setting, one is programmed as a central gateway (CGW) ECU and others control the powertrain, body, and chassis domains. They communicate CAN messages among each other. The CGW ECU is connected to the onboard diagnostic II (OBD-II) port using CAN and provides three CAN ports, and the other three ECUs are connected to the CGW ECU and assume the role of each domain. Each domain has one junction box as an expansion hub to which additional ECUs can be connected (Figures 4 and 5).

In an ECU, there are several typical attack surfaces such as the OBD-II port, clipping area, and junction box. In previous studies [5][6], a vehicle was attached and controlled by physical intrusion. To reproduce such attacks, we equipped PASTA with ECUs.

The simulated vehicle in PASTA is a simple one that reproduces the above three domains. To simulate physical characteristics such as weight and acceleration, PASTA and a driving simulator can be connected by CAN. As a result, the communication of the in-vehicle network in PASTA shows similar characteristics to an actual vehicle.

We equipped the OBD-II port, junction box, and area for wiretapping with clips as typical attack surfaces. Through these attack surfaces, attacks, such as wiretapping CAN messages flowing between ECUs and injection of malicious messages, can be reproduced.

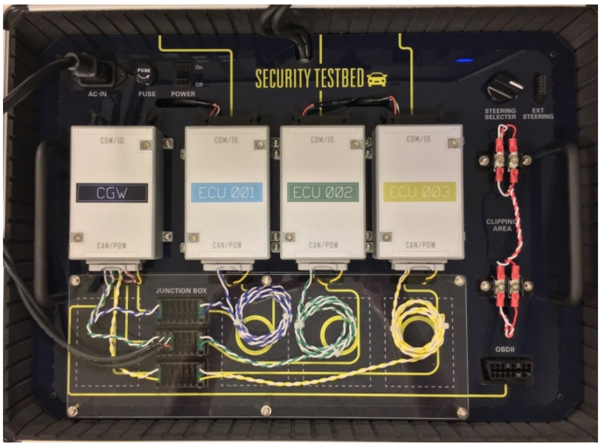

Figure 4. Embedded ECUs in PASTA. Leftmost ECU is central gateway (CGW), and other three on right control powertrain, body, and chassis domains

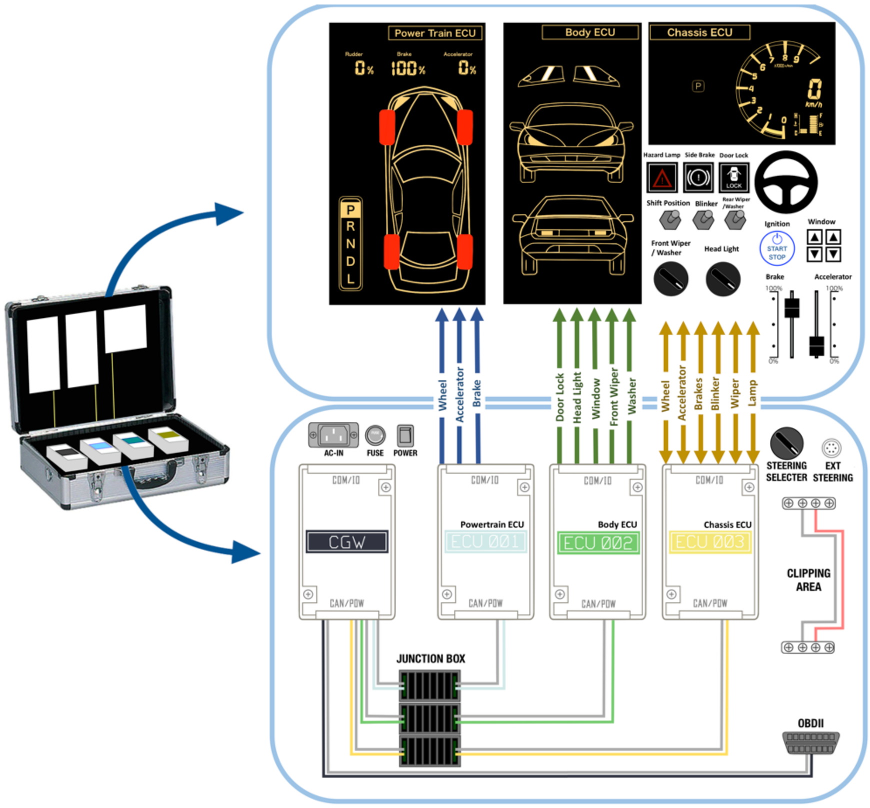

Figure 5. Structure of PASTA

3.2.2 Feature 2: Visualizing vehicle components

To achieve portability and safety with PASTA, vehicle behavior is displayed on the software-simulation screen. Portability is improved because there is no full-sized actuator. Even if the vehicle exhibits unexpected behavior when reproducing cyberattacks, the user remains safe. Details are described below.

Status panels in PASTA

In the upper half of the attaché case, three panels display the vehicle characteristics, e.g., speed, angle of tires, shift position, and status of headlight and brake lights, (Figure 6). The behavior of the simulated vehicle, which is affected by CAN messages travelling through the physical CAN bus, is displayed. This means that PASTA can display whether applied security technology is effective. If the simulated attack succeeds, the vehicle displayed on the panel operates abnormally.

Visualization of vehicle behavior by using scale model of vehicle and driving simulator

A 1/10-scale model of a vehicle (Figure 7) is suitable as an actuator. This scale model is connected to the testbed by Bluetooth® and receives the command of the actuator from the testbed. Even though it is compact, it is equipped with wipers, headlights, brake lights, blinkers, etc.

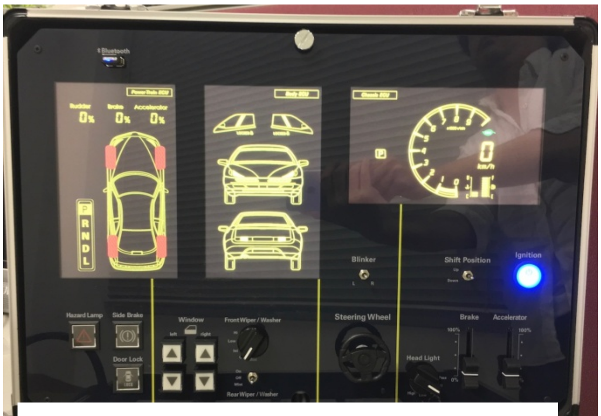

Figure 6. Panels on upper side of PASTA displaying vehicle status

that operate according to the commands. PASTA is also able to connect to a driving simulator (Figure 8). Since PASTA is highly flexible, when a CAN connection is available, an actual steering wheel or pedal can be used in the simulation.

Portability and Safety

PASTA is so compact that all the components can be stowed in one attaché case (Figure 2). This makes it possible to easily carry it and demonstrate research results in various places. Furthermore, because it is compact, it can be used as an educational tool of vehicle security even in small spaces. Since there is no full-sized actuator and the behavior can be confirmed with the software vehicle simulator, even if the vehicle behaves unexpectedly when an attack is reproduced, users remain safe because the vehicle is a simulation.

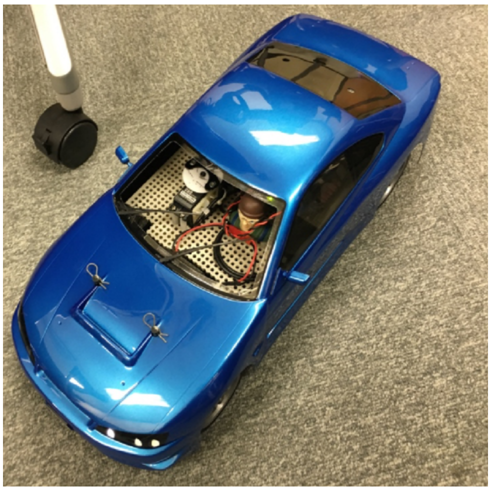

Figure 7. 1/10-scal model of vehicle connected to PASTA by Bluetooth®

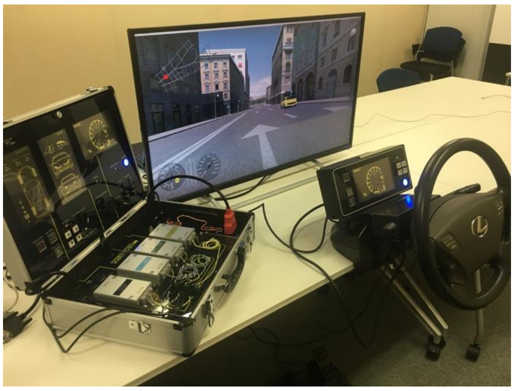

Figure 8. Driving simulator connected to PASTA

4. Considerations

Acceleration of security research

PASTA uses non-proprietary technology and visualizes vehicle behavior from in-vehicle network messages. This testbed is expected to provide a new platform for vehicle security, which will accelerate research and development.

When conducting experiments assuming a specific type of vehicle, it is difficult to completely release the research results. This is because OEMs tend to keep communication between the ECUs in the vehicle, the topology of the in-vehicle network, and countermeasures against expected attacks confidential. Once researchers release their activities, anyone can determine the vulnerabilities of a specific vehicle and confidential specifications and intellectual property. It is also difficult for OEMs to handle vulnerabilities rapidly because vehicle owners have to bring back their vehicles to the car dealers or automobile factory for repairs. If a research group releases vulnerabilities without the agreement of an OEM, the OEM might take legal action against the group. Auto-ISAC was established to share information on vulnerabilities; however, the results of Auto-ISAC are disclosed to members only.

The current research environment is not sufficient in terms of versatility and verifiability. When a research group proposes new cybersecurity technology for vehicles, evaluation is often conducted not on an actual vehicle but on an evaluation system they built with commercial products because it is difficult for such groups to obtain cooperation from OEMs. Research groups without OEM support have evaluated their ideas on their own environments. However, it is difficult for third parties to follow up and review the validation of their ideas. This means that other research groups cannot easily confirm whether it is suitable for actual vehicles.

Providing standard development platform

PASTA is characterized by high flexibility because white-box ECUs and general CAN message are available. Our scale model of a vehicle provides a standard platform to reproduce and evaluate various cyberattacks.

The ability to simulate unspecified vehicles using white-box ECUs has another advantage. For example, if a security hole is found in a model that simulates a specific vehicle, the automaker's stock price may fall. For this reason, security holes may not be made public. However, there is no such situation if it is an unspecified vehicle, and security of many automakers will be improved by publishing security holes. As a result, using a common platform makes it easier for researchers to share knowledge and improve their skills.

Visualization of CAN communication results

We equipped PASTA with a monitor for displaying CAN messages. It is also equipped with a controller, so variables in the simulator can be directly changed. We also developed an external controller, which can be used in the same way as an internal one. By connecting with the software vehicle simulator, we can receive feedback from the simulator. For example, when malicious CAN messages are injected for attacking steering control, we can feel the rattling of the steering wheel.

Therefore, it is possible to conduct well-known attacks such as sniffing CAN messages and injecting malicious CAN messages. PASTA highlights the threat of cybersecurity issues and makes students and vehicle-industry engineers aware of them.

Educational use

We have been using PASTA for student cybersecurity exercises for vehicles. Students were able to demonstrate well-known attacks, such as sniffing CAN messages via the OBD-II port and injecting malicious messages, in a few days. Therefore, we believe that PASTA can be an effective teaching tool. Furthermore, since they could refer to the behavior of the scale model of a vehicle or software vehicle simulator, we could determine whether to carry out cybersecurity exercises regarding the safety of the researchers and students.

Issues with and future work for PASTA

There are still points to be improved in PASTA. First, the software vehicle simulator in PASTA is still immature. For example, if you step on an accelerator, it will reach a speed of 199 km/hour in a very short time. In that case, the CAN message flowing in the in-vehicle network in PASTA does not reflect the appropriate vehicle condition. Although this problem has been mitigated to some extent with the physical simulator, optimization of the software vehicle simulator is a future task.

Second, ECUs in the current version of PASTA can handle only CAN as a protocol. Since CAN is currently the most popular communication protocol for ECUs, PASTA can address most of the currently expected cyberattacks via this protocol. However, simulation assuming cyberattacks using protocols other than CAN cannot be reproduced yet. Therefore, we plan to develop ECUs that correspond to protocols such as CAN with Flexible Data-Rate, Local Interconnect Network, Media Oriented Systems Transport, or Ethernet.

Third, preparation for various attack surfaces is required. PASTA has attack surfaces to reproduce the physical intrusion via the OBD-II port or tapped CAN cable. However, with the popularity of wireless connected technology for vehicles in recent years, it is necessary to implement Wi-Fi, Bluetooth, or cellular networks. Past studies succeeded in controlling a vehicle from wireless attack surfaces [7][8]. Their results have attracted much attention since this threat is expected to affect many connected vehicles; thus, measurement and evaluations on cyberattacks via wireless attack surfaces are necessary. By equipping PASTA with a wireless environment, we will be able to create an environment suitable for wireless cyberattacks.

Finally, we are currently using our own software architecture as the implementation environment of the ECUs. Automotive Open System Architecture has recently been widely used as an open and standardized software architecture, and it is required to follow this.

5. Conclusion

In this paper, we outlined the introduction and possibilities of our vehicle cybersecurity testbed, PASTA. We developed an example of one form of PASTA for proof of concept. Any portable testbed with open specifications and to which security techniques can be freely applied can be called PASTA.

The primary objective of PASTA development is to provide a development and evaluation environment for vehicle cybersecurity technology, and the secondary objective is to visualize various internal operations. As a result, PASTA is also helpful for education. It is expected to become widely used due to its portability and affordability. We hope that this testbed will be effective as a platform for vehicle cybersecurity research.

References

[1] K. Fischer, “High Assurance Cyber Military Systems (HACMS),” escar USA2013, 2013.

[2] C. Miller, C. Valasek, “Car Hacking: For Poories,” SyScan2014, 2014.

[3] J. Munera, J. M. de Fuentes, A. I. González-Tablas, “Towards a comparable evaluation for VANET protocols: NS-2 experiments builder assistant and extensible test bed,” escar Europe 2011, 2011.

[4] Auto-ISAC website, https://www.automotiveisac.com

[5] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, “Experimental Security Analysis of a Modern Automobile,” IEEE Symposium on Security and Privacy, 2010.

[6] C. Valasek, C. Miller, “Adventures in Automotive Networks and Control Unit”, http://www. ioactive.com/pdfs/IOActive_Adventures_in_ Automotive_Networks_and_Control_Units.pdf, 2014.

[7] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, “Comprehensive experimental analyses of automotive attack surfaces,” the 20th USENIX Security Symposium, 2011.

[8] C. Miller, C. Valasek, “Remote Exploitation of an Unaltered Passenger Vehicle,” http://illmatics.com/Remote%20Car%20Hacking.pdf, 2015.