Analysis of Cybersecurity Systems in the Automotive Sector

by
Roberto Gesteira Miñarro

supervised by
Dr. Gregorio Ignacio López López
Dr. Rafael Palacios Hielscher

A document submitted for the degree of
Doctor of Philosophy
at
ICAI School of Engineering
Universidad Pontificia Comillas

Madrid, 2025

To the people who love me. Especially,
to my beloved partner, my family and my friends.

«Don't waste your time or time will waste you.»
– Muse

«Life is what happens while you are busy making other plans.»
– John Lennon

«La vida se nos va tan rápido, no hay tiempo de sentir el vértigo.»
– Fito & Fitipaldis

«Y no me importa si es mentira todo, lo que vivimos nadie nos lo va a quitar.»
– Arde Bogotá

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Roberto Gesteira Miñarro
Madrid, 2025

Agradecimientos

A mis directores de tesis doctoral: Gregorio López y Rafael Palacios, por su guía constante, su exigencia intelectual y su apoyo humano durante estos años. Su ejemplo ha sido fundamental para mi crecimiento académico y personal.

A Carmen, por la gran paciencia que ha tenido durante el desarrollo de la tesis, su apoyo en los momentos malos y su alegría en los momentos buenos, por su ayuda e interés en todo momento, y por buscar siempre lo mejor para mí.

A Ana, Miguel y Miguelito, por confiar en mí, por tenerme siempre en cuenta, y por ayudarme y apoyarme con mis intereses.

A mi madre y mis abuelos; y a mi padre, mis hermanos y Bea; por su apoyo incondicional y por haberme hecho llegar donde estoy ahora.

A mis amigos de toda la vida: Ramiro y Alberto, por estar siempre ahí cuando los necesité.

No quiero terminar esta sección de agradecimientos sin mencionar a estas personas, las cuales también han influido en el desarrollo de esta tesis:

- A mis amigos de ICAI (Juan, Pelayo, Liher, Alejandro, Emmanuel, Mario, Javi, Fo, Ana, Jorge, Nacho, Mery, Nico, Diego, Carmen, Jaime), y profesores (Mario Castro, Javier Matanza, Pablo Sánchez, Antonio Pérez, Carlos Rodríguez-Morcillo, José Luis Rodríguez Marrero, Javier Jarauta, David Contreras, Atilano), y en especial al Departamento de Matemática Aplicada.
- A mis compañeros del IIT (Juanma, Paulo, Claudia, Néstor, Jaime, Andrea, Farid, María, Geovanny, Léo, Jorge, Nacho, Diego, Fran, David, Leandro).
- A mis compañeros de KU Leuven COSIC (Peter, Emil, Varesh, Kevin, Jakob, Enrique, Can, Theresa, Felix, Oliver, Riccardo, Krijn, Rafa, Mariana, Martin, Jonas, Péla), y en especial a Takahito Yoshizawa por su ejemplo y dedicación al trabajar conmigo en V2X; a Robin Jadoul por ser mi contacto inicial en COSIC y por ser un referente en competiciones de CTF de criptografía; y a Bart Preneel por alojarme en COSIC para mi estancia internacional.
- A mis amigos de ICAI Cyber Team (Lucas, Alberto, Álvaro, Fran, Carlos, Teresa, David), y en especial a Nacho.
- A mis amigos, los *hackers* (Carlos, Jorge, Dani, Yago, Marc, Enrique, Nacho, Isaac, Jairo, Marc, Joan, Jose, Víctor, Juan, Gonzalo, Bea, Laura).
- A mis amigos de thehackerscrew (Moriarty, Aali, ItayB, KLPP, bubu).
- A mis amigos de judo (Pablo, Carlos, Ignacio, José Luis, Antonio, Miguel, Gonzalo, Juan, Javi), y en especial al *sensei* Javier.
- A mis amigos Joserra y José Ramón.
- A mi amigo Don Ángel.
- A los creadores de retos de Hack The Box, CryptoHack y competiciones de CTF en las que he participado.

Abstract

The automotive industry has been a prime target for cybercriminals for decades, with attacks becoming increasingly sophisticated as vehicles integrate advanced digital technologies. Modern vehicles present an extensive attack surface due to the growing number of interconnected electronic components and wireless communication features. While these technologies improve connectivity, automation, and comfort, they also introduce new vulnerabilities that can be exploited by attackers. In response, new standards and regulations require manufacturers to implement robust cybersecurity measures to obtain required certifications. This dissertation explores the attack surface of modern vehicles by analyzing several critical systems and technologies.

First, a comprehensive study of wireless communication technologies is presented, highlighting existing research, known vulnerabilities, and potential countermeasures, while identifying key research gaps that require further investigation. In addition, background about Bluetooth and OBD-II dongles is presented; as well as concepts related to Vehicle-to-Everything (V2X) communications.

Second, Remote Keyless Entry (RKE) protocols are examined, with the identification and analysis of a vulnerability that enables key fob cloning through a black-box reverse-engineering methodology. This approach relies solely on open-source tools and commercially available hardware, avoiding the need for expensive firmware extraction tools. This methodology is applied to eight protocols from different manufacturers, trying to determine the protocol structure. As a result, a detailed analysis of eight protocols from different manufacturers is shown and they are compared from a security perspective, with one of them being totally broken.

Third, the attack surface exposed by OBD-II dongles is addressed through the design of a research and demonstration platform for testing these devices, accompanied by the software tool `pwnobd`. This platform enables vulnerability analysis and penetration testing without requiring direct access to a vehicle. Several diagnostic devices were evaluated, including a focused case study that illustrates the benefits of this platform for security researchers.

Finally, V2X communication protocols are analyzed with respect to ETSI Intelligent Transport System (ITS) standards, focusing on Cooperative Awareness Messages (CAM), Decentralized Environmental Notification Messages (DENM), and pseudonym mechanisms. Multiple vulnerabilities were identified, enabling replay attacks, spoofing attacks, Sybil attacks, and grayhole attacks. In addition, attack scenarios are developed and quantified through simulations using Eclipse SUMO, highlighting risks during the coexistence of legacy and V2X-enabled vehicles. Mitigations are proposed to address these issues, underscoring the need for stronger security measures to preserve safety and trust in future intelligent transportation systems.

Resumen

El sector del automóvil ha sido un objetivo principal para los cibercriminales durante décadas, con ataques cada vez más sofisticados a medida que los vehículos integran tecnologías digitales avanzadas. Los vehículos modernos presentan una amplia superficie de ataque debido al creciente número de componentes electrónicos interconectados y funciones de comunicación inalámbrica. Si bien estas tecnologías mejoran la conectividad, la automatización y la comodidad, también introducen nuevas vulnerabilidades que pueden ser explotadas por atacantes. En respuesta, se han introducido nuevas normativas y regulaciones que obligan a los fabricantes a implementar medidas de ciberseguridad robustas y obtener las certificaciones requeridas. Esta tesis explora la superficie de ataque de los vehículos modernos mediante el análisis de varios sistemas y tecnologías críticas.

En primer lugar, se presenta un estudio exhaustivo de las tecnologías de comunicación inalámbrica, destacando la investigación existente, las vulnerabilidades conocidas y las posibles contramedidas, a la vez que se identifican vacíos clave que requieren mayor investigación. Además, se ofrece un contexto sobre Bluetooth y los *dongles* OBD-II; así como conceptos relacionados con las comunicaciones *Vehicle-to-Everything* (V2X).

En segundo lugar, se examinan los protocolos de *Remote Keyless Entry* (RKE), con la identificación y el análisis de una vulnerabilidad que permite clonar mandos con una metodología de ingeniería inversa de caja negra. Este enfoque se basa exclusivamente en herramientas de código abierto y *hardware* comercial, evitando la necesidad de herramientas costosas para la extracción de *firmware*. La metodología se aplica a ocho protocolos de diferentes fabricantes, con el objetivo de determinar su estructura. Como resultado, se presenta un análisis detallado de los ocho protocolos y se comparan desde una perspectiva de seguridad, demostrando que uno de ellos puede ser completamente comprometido.

En tercer lugar, se aborda la superficie de ataque expuesta por los *dongles* OBD-II mediante el diseño de una plataforma de investigación y demostración para probar estos dispositivos, acompañada de la herramienta de *software* `pwnobd`. Esta plataforma permite realizar análisis de vulnerabilidades y pruebas de penetración sin necesidad de acceder directamente a un vehículo. Se evaluaron varios dispositivos de diagnóstico, incluyendo un caso de estudio específico que ilustra los beneficios de esta plataforma para los investigadores en ciberseguridad.

Finalmente, se analizan los protocolos de comunicación V2X en relación con los estándares de *Intelligent Transportation Systems* (ITS) del ETSI, centrándose en los *Cooperative Awareness Messages* (CAM), los *Decentralized Environmental Notification Messages* (DENM) y los mecanismos de pseudónimos. Se identificaron múltiples vulnerabilidades que permiten ataques de *replay*, *spoofing*, Sybil y *grayhole*. Además, se desarrollaron y cuantificaron escenarios de ataque mediante simulaciones con Eclipse SUMO, destacando los riesgos durante la coexistencia de vehículos heredados y vehículos con V2X habilitado. Se proponen medidas de mitigación para abordar estos problemas, subrayando la necesidad de fortalecer las medidas de seguridad con el fin de preservar la seguridad y la confianza en los futuros sistemas de transporte inteligente.

Contents

List of Figures

List of Tables

List of Acronyms

ACK	Acknowledgement
ADB	Android Debug Bridge
AES	Advanced Encryption Standard
AI	Artificial Intelligence
API	Application Programming Interface
ASK	Amplitude Shift Keying
BKM	Butterfly Key Mechanism
BLE	Bluetooth Low Energy
BSM	Basic Safety Messages
C-ITS	Cooperative ITS
CA	Cooperative Awareness
CAM	Cooperative Awareness Message
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
CTF	Capture The Flag
CVE	Common Vulnerabilities and Exposure
DCC	De-centralized Congestion Control
DENM	Decentralized Environmental Notification Message
DNS	Domain Name System
DoS	Denial of Service
DSRC	Dedicated Short Range Communication
ECDSA	Elliptic Curve Digital Signature Algorithm
ECU	Electronic Control Unit
ETSI	European Telecommunications Standards Institute
FPGA	Field-Programmable Gate Array
FSK	Frequency Shift Keying
GATT	Generic ATTribute Profile
GPS	Global Positioning System
HCI	Host Controller Interface
I^2C	Inter-Integrated Circuit
I2V	Infrastructure-to-Vehicle
ID	Identifier
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IFAL	Issue First Activate Later

IoT	Internet of Things
ITS	Intelligent Transportation Systems
ITS-S	ITS Stations
JTAG	Joint Test Action Group
LDM	Local Dynamic Map
LFSR	Linear Feedback Shift Register
LTE	Long Term Evolution
M2M	Machine-to-Machine
MAC	Medium Access Control
NLFSR	Non-Linear Feedback Shift Register
OBD	On-Board Diagnostics
OBU	On-Board Unit
OOK	On-Off Keying
P2PCD	Peer-to-Peer Certificate Distribution
PKES	Passive Keyless Entry and Start
PKI	Public-Key Infrastructure
PQC	Post-Quantum Cryptography
PRNG	Pseudo Random Number Generator
RFID	Radio Frequency Identification
RKE	Remote Keyless Entry
RSSI	Received Signal Strength Indicator
RSU	Road Side Unit
RTT	Round-Trip Time
SAE	Society of Automobile Engineers
SCMS	Security Credential Management System
SDO	Standard Defining Organization
SDR	Software-Defined Radio
SPI	Serial Peripheral Interface
SWD	Single Wire Debug
SUMO	Simulation for Urban MObility
TPMS	Tire Pressure Monitoring System
TraCI	Traffic Control Interface
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VAM	VRU Awareness Message
VANET	Vehicular Ad-Hoc Network
VRU	Vulnerable Road User
WAVE	Wireless Access in Vehicle Environment
XTEA	eXtended Tiny Encryption Algorithm
ZKP	Zero-Knowledge Proof

Chapter 1

Introduction

This chapter discusses the motivation, and scope and objectives of this thesis. Finally, the overall structure of the thesis is presented.

1.1 Motivation

Modern vehicles are designed with many features to provide human safety and comfort. However, adding more functionalities can lead to security issues, since there are more assets to protect and more potential vulnerabilities to detect. As a result, new regulations have come into force in the automotive industry, and manufacturers must take cybersecurity into account to pass certain certifications (UN Regulation No. 155, 156) [UNE21a; UNE21b] before selling their products [GHM23]. With the rise of autonomous vehicles, connected vehicles, shared vehicles, and Vehicle-to-Everything communications (V2X), an increase in research projects in this field is expected.

The automotive industry has been a target for cybercriminals for decades. There are many resources related to automotive cybersecurity (also known as "car hacking" in the hacker community). Nevertheless, most of the knowledge is kept in cybersecurity conferences rather than in academic papers. It is common to see talks about car hacking in cybersecurity conferences such as Black Hat or DEF CON. In fact, every year there is a Car Hacking Village [Vil] (associated to DEF CON) that joins several researchers on this topic. It is worth mentioning the CyberAuto Challenge and the CyberTruck Challenge [DS24] as challenge-based models for developing talent and building community among cybersecurity researchers and industry. In addition, there is the Automotive Security Research Group (ASRG) [ASR]; or VicOne [Vica], which is a company that focuses on automotive cybersecurity and sponsors the Pwn2Own Automotive competition [Vicb], where hackers from all over the world try to compromise some of the proposed targets for fame and bounties.

In this topic, private-industry researchers are more advanced than academic researchers. In fact, many of the attacks and techniques covered in this thesis dissertation are already implemented in hardware devices that can be purchased in underground forums in the Dark Web, probably illegal.

The peak of car hacking came when Miller and Valasek showed in Black Hat USA 2015 [Dro15] how to compromise a 2014 Jeep Cherokee [VM18] using different attack vectors. The authors also wrote a white paper [MV15], which contains almost the same information, but in a more formal register. Although there are papers on automotive cybersecurity before, this milestone triggered a lot of research and cybersecurity awareness regarding vehicles. The same year, Samy Kamkar presented RollJam at DEF CON 23 [Kam15], an attack to bypass rolling-code implementations on Remote Keyless Entry (RKE) systems.

In order to improve and test the security of existing and future vehicles, car manufacturers need to rely on research projects, to have an external perspective. This research is not only valuable for vehicle manufacturers, but also for insurance companies, since they need to estimate the insurance rates depending on different factors, including the cybersecurity level of a given vehicle. In fact, there are insurance companies that nowadays are not willing to handle some car models due to their high rate of car theft [Val23]. Although modern entry systems have several mitigation features, news about car theft are very frequent. Cybercriminals always find ways to exploit vulnerabilities and bypass these protection mechanisms. This is the reason why countries such as Canada have recently prohibited the use of hacking devices like Flipper Zero [PN24], which are known to be used for car hacking.

The attack surface of a vehicle is very wide, and keeps growing with the advances in technology. A car is composed of a large number of Electronic Control Units (ECUs). These components are sensors and actuators that connect the car with the external environment. Each ECU has a specific functionality, and all of them are connected with each other via serial buses. The communication between the different ECUs is handled by the Controller Area Network (CAN) protocol [SHKL22]. The CAN bus uses an old protocol that was not designed with cybersecurity in mind. For instance, there is no encryption, no authentication, and the network has a bus topology. As a result, the CAN bus is a juicy target for cybercriminals to compromise a vehicle, because they can potentially eavesdrop CAN messages from the ECUs and send arbitrary CAN messages or commands to any ECU. These issues are already addressed in [BSJ18], as well as several ideas for an Intrusion Detection System (IDS) to mitigate attacks in [Wu+20] or [YZOB19].

Although the CAN bus is accessible through the On-Board Diagnostics port (OBD-II) for diagnosis purposes in a car workshop [BSJ18], it is also connected to the infotainment system (information and entertainment systems) in modern vehicles, which poses a higher risk and attack surface for cybercriminals to compromise a vehicle. CAN bus vulnerabilities are not likely to be corrected because communication between ECUs must be extremely fast, as human safety might be affected. Therefore, the CAN bus must be a fast network where latency is negligible. Because of that, there is no encryption or message authentication, to minimize processing times on the ECU side (as well as power consumption, among other characteristics). Therefore, it is the most critical target for attackers in order to compromise a vehicle. In this context, thieves have found a technique known as CAN injection [Tin23] in order to get access to the CAN bus. They take a twisted

pair cable inside the headlights and use it to connect directly to the CAN bus, so that they can read and inject arbitrary CAN messages [Pal23].

With the advances in artificial intelligence, image processing and cellular communications, among other technologies, new concepts have appeared: autonomous vehicles, vehicle-to-vehicle (V2V) communications, vehicle-to-infrastructure (V2I) communications, and, in general, Vehicle-to-Everything (V2X) communications. These new advances in the automotive sector are being designed nowadays, with cybersecurity and human safety in mind. It is important to keep these communications as fast as possible, minimizing latency, while making them secure and private. There are already papers that cover these topics in depth, such as [GHM23], [Cho+20], [PS15] or [SYZ22]. More specific security issues in V2X communications can be found in [YP19].

The consequences of performing a cyberattack on a vehicle can result in denial of service (DoS), car theft, personal-asset theft, information disclosure, cyber-physical damages, or even remote control, among others. Thus, it is crucial to protect the automotive ecosystem in order not to affect human safety.

1.2 Scope and objectives

The main objective of this thesis is to develop novel and flexible methodologies and tools to analyze the level of cybersecurity of nowadays and future vehicles and their related technologies. In particular, these are the individual objectives pursued on this thesis:

1. Analyze existing cyberattacks and search for new attack vectors and vulnerabilities.
2. Build a methodology for analyzing automotive subsystems from a black-box perspective.
3. Develop tools for auditing automotive subsystems.
4. Find security gaps in V2X communications protocols and standards for connected vehicles.

With these objectives, this thesis aims to cover some of the gaps identified in the state of the art; namely, reverse engineering, cryptography, mitigations, software security and V2X communications.

1.3 Thesis outline

The structure and content of each chapter are briefly described below. Figure 1.1 shows the main topics that will be covered on the thesis, which follows the interconnection of the chapters. The main contributions of this thesis will cover wireless attacks (RKE/PKES and Bluetooth), physical attacks (OBD-II) and protocol attacks (V2X).

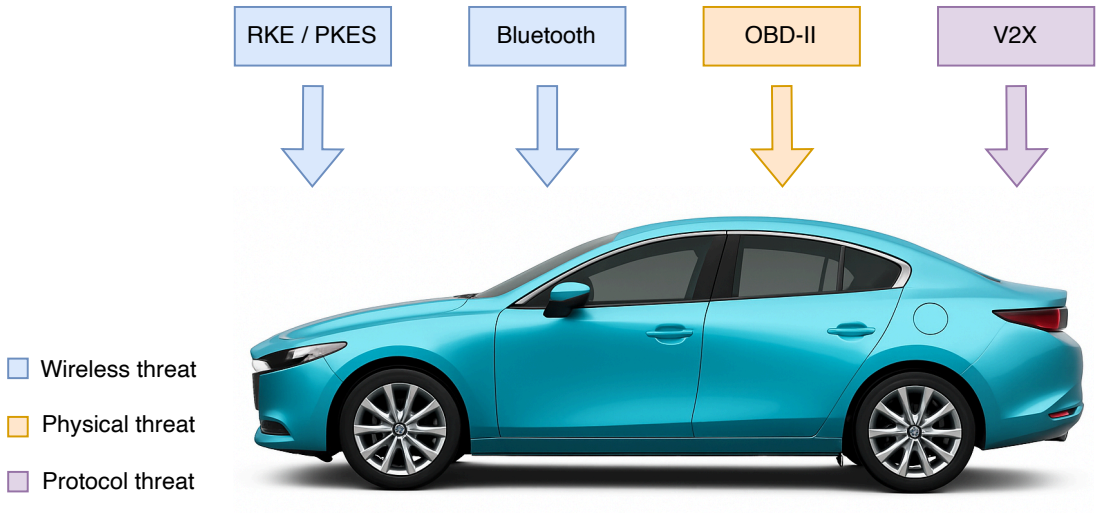


Figure 1.1. Topics covered on this thesis.

- Chapter 2 aims to provide a general background and state-of-the-art review for the rest of the chapters.
- Chapter 3 describes a methodology to analyze the security level of keyless entry systems using reverse engineering on radiofrequency protocols employed for the communication between the key fob and the vehicle.
- Chapter 4 introduces the issues of using OBD-II dongles that connect to a mobile application via Bluetooth, and presents a tool called `pwnobd` to audit these devices.
- Chapter 5 analyzes the current state of V2X communication protocols and shows some security gaps found in the ETSI Intelligent Transportation System (ITS) standards. In addition, attack scenarios are presented, along with simulations and solutions.
- Chapter 6 is the last chapter of the thesis. First, the conclusions and contributions are discussed. Then, promising lines of future work in the field of automotive cybersecurity are presented. Finally, the publications of the thesis are listed.

This thesis dissertation covers material from the following publications:

- Roberto Gesteira-Miñarro, Gregorio López, and Rafael Palacios. "Revisiting Wireless Cyberattacks on Vehicles". *Sensors* 25.8 (Apr. 2025). ISSN: 1424-8220. DOI: 10.3390/s25082605.
- Roberto Gesteira-Miñarro, Gregorio López, and Rafael Palacios. "Clonable key fobs: Analyzing and breaking RKE protocols". *International Journal of Information Security* 24.3 (May 2025). ISSN: 1615-5262. DOI: 10.1007/s10207-025-01063-7.

- Roberto Gesteira-Miñarro, Ignacio Gutiérrez, Rafael Palacios, and Gregorio López. "pwnobd: Offensive Cybersecurity Toolkit for Vulnerability Analysis and Penetration Testing of OBD-II Devices". *IEEE Access* 13 (July 2025), pp. 126925–126934. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2025.3589867.
- [*Under review*] Roberto Gesteira-Miñarro, Takahito Yoshizawa, Gregorio López, and Rafael Palacios. "Highway to Hack – Security Gaps in ETSI ITS Standards". *Computer Standards & Interfaces* (2025). ISSN: 0920-5489.

Chapter 2

State of the Art

This chapter presents background and related works regarding wireless cyberattacks on vehicles, including keyless entry systems (RKE and PKES), immobilizers, and other wireless technologies (TPMS, Bluetooth and GPS). Plus, this chapter shows other technologies such as Bluetooth OBD-II dongles and V2X communications. Finally, research gaps are identified based on this state-of-the-art review.

2.1 Remote Keyless Entry

The keyless entry system of a car allows users to lock and unlock their vehicles, among other actions such as opening the trunk or starting up the engine remotely. The Remote Keyless Entry (RKE) system is simple from the user's point of view, since they only need to press a button on their key fob, and the action is performed automatically. From the technical point of view, a digital signal is sent by the key fob and the car analyzes that signal and executes the requested action.

RKE systems are vulnerable to replay attacks and jamming by construction, due to how they work. Some tools to perform these attacks are presented in [IHOD19], but these are *script-kiddie* techniques, because the requirement is only to have the proper tools. In [GA22] a reverse-engineering approach is shown to analyze garage door openers. This perspective is more insightful because all the protocol details are discovered, so that the attacks can be understood in a deeper way. Similar resources to learn about reverse engineering in radio frequency protocols can be found in [Oss] and [Ken24].

RKE signals are usually modulated with digital modulations. RKE signals are usually modulated using digital schemes such as Amplitude Shift Keying, particularly On-Off Keying (ASK/OOK); or Frequency Shift Keying, especially with two carrier frequencies (2FSK). In most cases, the signals start with a synchronization sequence and also use Manchester coding as a channel encoding, to prevent synchronization errors on the receiver. Furthermore, some key fobs will send the same information several times to ensure that the receiver gets the signal, as a redundancy mechanism.

Most of the protocols used for keyless entry are proprietary and closed-source. However, this security-by-obscurity principle is usually beaten with reverse engi-

neering or information leaks in underground forums. This is the case of KeeLoq, which is a block cipher owned by Microchip [End10]. After the confidential specifications were leaked, a lot of research papers emerged and KeeLoq was proven to be a weak cipher, with various cryptographic and side-channel attacks. Other known keyless entry ciphers are Hitag2 [VGB12] or AUT64 [HGO18].

2.1.1 Replay attacks

A vehicle will unlock whenever a valid signal is received, thus showing that RKE systems have an intrinsic vulnerability, because it does not matter the device that sends the signal. As a result, if an attacker manages to capture a valid signal, they might be able to unlock the car by sending the captured signal [GA22], which is known as a replay attack.

Rolling codes were introduced to mitigate replay attacks in RKE systems. These codes can only be used once. The intuition is that the key fob and the vehicle have a Pseudo Random Number Generator (PRNG) initialized with the same seed, so that the key fob sends the next step of the algorithm and the vehicle can match the code within a list of valid codes. Consequently, once a code is used, it is marked as invalid to prevent replay attacks.

Other rolling-code implementations make use of symmetric ciphers like Hitag2 or KeeLoq. With this approach, the key fob holds a counter that is increased on every button press. This counter is encrypted along with other fields such as key identifiers and sent along with the encoded command. The vehicle is able to decrypt the rolling code and determine if the counter is valid or not and advance it accordingly [MK09].

Regardless of the type of rolling-code implementation, it must be indistinguishable and not predictable for an adversary. Namely, given a rolling code, it must be impossible to tell if it comes from a PRNG or a cipher, even with multiple samples.

Figure 2.1 illustrates the way rolling codes work, using a PRNG-based approach and dummy numbers that mimic rolling codes. The list of numbers at the right represents the PRNG status of the key fob, saying that code 1234 is already used, 2345 is the next PRNG output and the rest are future outputs. To the left, there is a list of rolling codes: 1234 is discarded because it was already used before; 2345, 3456, 4567 are valid rolling codes; and 5678 will be added later. Following the steps in the figure, (1) the user presses a button on the key fob to open the car. The corresponding signal holds rolling code 2345. Then, (2) the vehicle checks if the rolling code is within the list of valid codes. If not, the vehicle must ignore the request. If the code is valid, it is marked as invalid (just like 1234) and (3) the action is performed. After that, (4) the vehicle adds a new valid code to the list, namely 5678. A similar procedure can be implemented for cipher-based rolling codes. Instead of having a list of valid codes, the vehicle needs to decrypt the rolling code and tell if the encoded number is greater than its current counter. If so, the action is performed; if not, it means that the signal held a past counter value, so if must be discarded.

In PRNG-based rolling codes, the list of valid rolling codes must be sufficiently large. The aim is to preserve synchronization with the key fob, because the user may press the button by accident and waste some rolling codes that the vehicle will never receive. Otherwise, the next PRNG output of the key fob would not be in the list and the car will not perform the requested action.

Nevertheless, there are still ways to perform replay attacks. The limitation is that the vehicle must not receive the code. Therefore, if an attacker captures a signal with a valid code from the key fob and the car is not near, they will be able to replay the signal and the car will behave normally because the code is treated as valid.

In some cases, a brute-force approach might be useful. There was a talk at DEF CON 32 [Era24] about a specific type of programmable rolling codes, known as learning codes, that are only 20 bits long. The researcher showed how he was able to reverse-engineer the RKE protocol to find its structure and perform a 20-bit brute-force attack to unlock these cars.

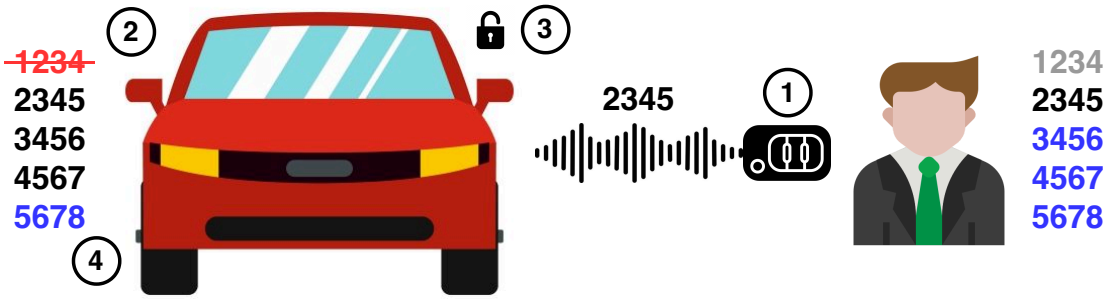


Figure 2.1. Operation of rolling codes.

2.1.2 RollJam attack

In relation to rolling codes, Samy Kamkar presented a new technique called RollJam [Kam15] in DEF CON 23 to attack rolling-code implementations in RKE systems. The aim of the attack is to obtain a set of valid codes that can be used to unlock the car later.

The attack involves using a jamming antenna (jammer) near the vehicle (i.e. under the vehicle to remain unnoticed) in order to generate interference in the working frequency band. As a result, the legitimate key fob will not work as expected because the car will not receive any signal. As shown in Figure 2.2, (1) the user will press the key fob buttons several times, but the car will not perform any action due to the interferences. Meanwhile, (2) the attacker can capture these failed rolling codes. The attacker must be next to the key fob to capture those signals, but far enough from the car to be outside the interference range. These codes (3) are valid because the car has not received any of them, so they are still in the list of valid codes (or hold a valid encrypted counter). Eventually, (4) the attacker should deactivate the jammer and use one of the first captured signals to unlock the car and make the victim think that the key fob is working again [Kam15]. Otherwise, the user might open the car with the physical key,

which ensures that all the captured codes have not being used and improves Samy Kamkar's approach [IHOD19]. These valid codes grant access to the interior of the vehicle and potentially steal personal belongings and information from the car owners and users.

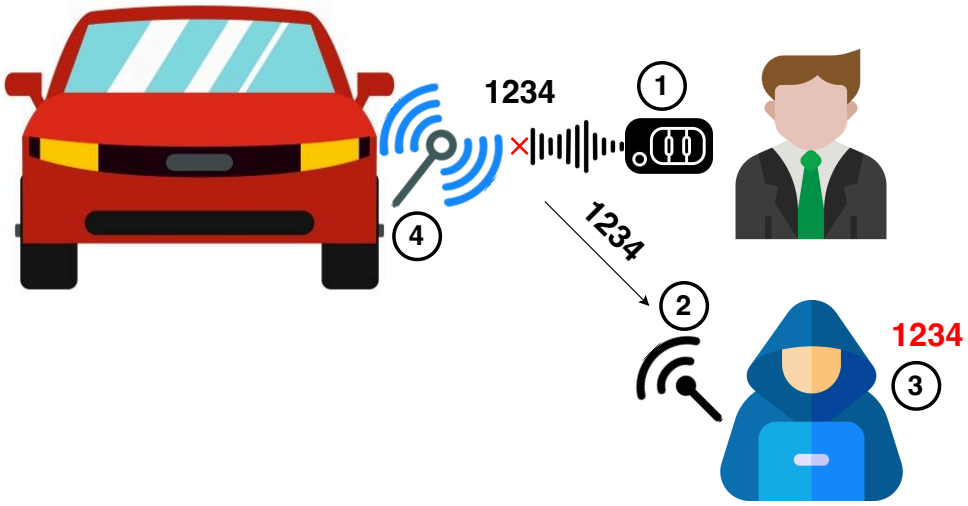


Figure 2.2. RollJam attack scenario on an RKE system.

Considering jamming techniques, in [BL16] it is discovered that RKE key fobs employ an envelope detector, which is much more vulnerable to pulsed electromagnetic interference than to continuous interference. The researchers show that the use of a synchronous detector mitigates the issue.

There are other rolling-code implementations that intend to protect against replay and RollJam attacks, for instance, in [GME17]. On the other hand, [PS22] proposes a protocol based on RSA signatures using a hash function to take the current date and the current time into account, as well as a random number generated with a PRNG using an increasing counter as a seed on both the key fob and the vehicle.

2.1.3 RollBack attack

Another vulnerability was found in rolling-code implementations, which gave birth to an attack known as RollBack [Csi+22], presented in Black Hat USA 2022. The researchers found that sending two or more already used and consecutive signals, the vehicle will resynchronize the key fob and perform the encoded action.

This attack is much more dangerous, powerful and easy to perform. The attacker just needs to capture two or more signals that the vehicle has received (it does not matter the type of action). The researchers showed a proof-of-concept video that proves how they were able to reuse these consecutive signals multiple times to unlock the vehicle. Nevertheless, not every vehicle brand is vulnerable to this attack.

2.2 Passive Keyless Entry and Start

In Passive Keyless Entry and Start (PKES) systems, the user is not required to press any button on the key fob. They just need to approach the vehicle and the doors will unlock after a bidirectional communication between the key fob and the vehicle [JCL20]. Figure 2.3 shows the differences between RKE and PKES message flows.

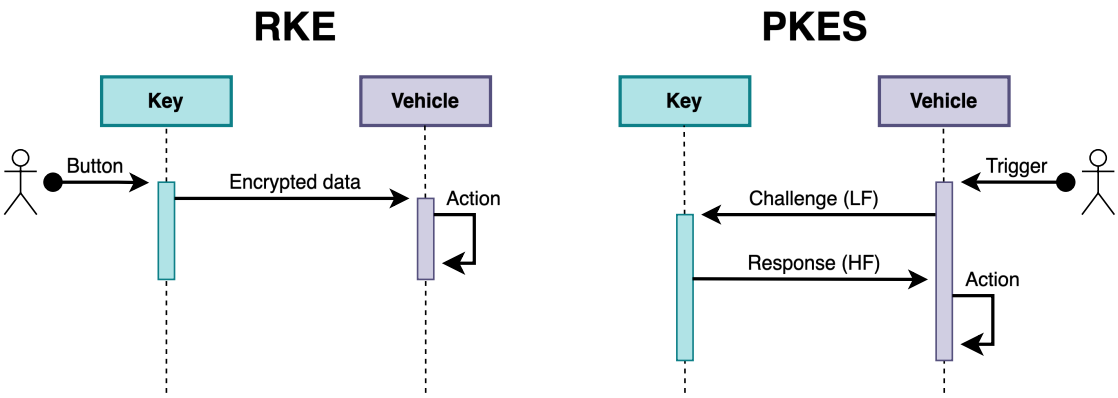


Figure 2.3. RKE and PKES message flows.

The "Trigger" signal in PKES systems can be different depending on the manufacturer. There are vehicles that require the user to touch the door handle, whereas other vehicles send a beacon signal periodically and wait until a key fob responds with an ACK. In both cases, once the "Trigger" action is completed, the vehicle sends a cryptographic challenge which only the legitimate key fob configured in the car is supposed to solve. If the response from the key fob is correct, then the vehicle will perform the required action.

In PKES systems, the process of starting the engine is similar, there is no need to enter the key to switch on the engine. The car verifies that the key fob is inside the vehicle and the engine starts after the user has pressed a "Start" button (which is the "Trigger" signal in Figure 2.3).

2.2.1 Relay attacks

PKES systems are vulnerable to relay attacks due to the way the protocol works. The attack scenario requires two malicious attackers, one near the key fob and one next to the car. As Figure 2.4 shows, (1) the attacker who is next to the car causes the "Trigger" signal and (2) relays the "Challenge" signal to the attacker that is next to the key. Then, (3) the "Challenge" signal is amplified and the key fob responds (as if the signal was produced by the car). Finally, (4) the "Response" signal is relayed to the attacker next to the vehicle and amplified, so that (5) the vehicle unlocks the doors since the response message is correct. Furthermore, the attackers can potentially start the engine because the process is mostly the same. Figure 2.4 illustrates how the attack is performed [AM03].

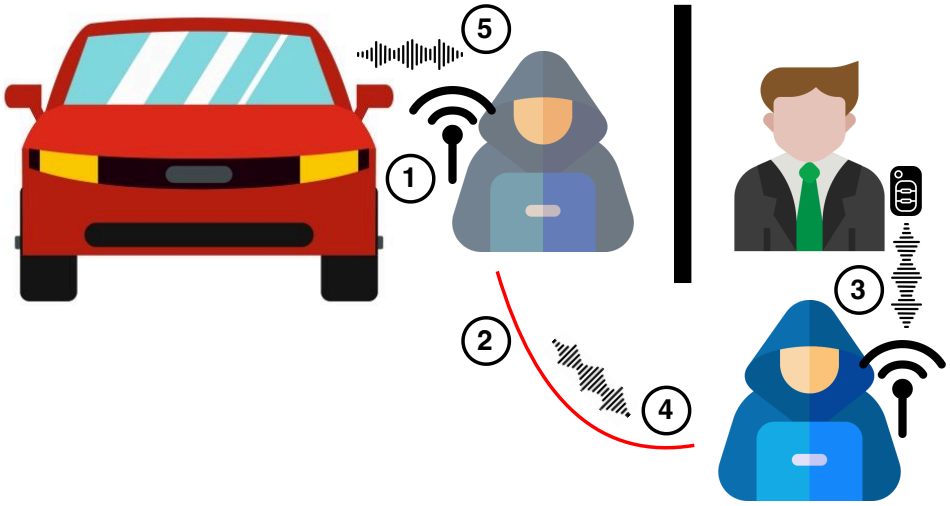


Figure 2.4. Relay attack scenario on a PKES system.

In fact, relay attacks on PKES systems are more dangerous than attacks on RKE systems because the attackers can potentially steal the car and go away, since PKES systems are also used to start the engine. The devices required for the attack are not expensive, and the communication between the attackers can be either wired or wireless [FDC11].

Regarding mitigations to relay attacks on PKES systems, in [WLZ19] the researchers propose the use of Bluetooth Low Energy (BLE) and several parameters like RSSI, RTT, GPS or Wi-Fi access-point lists to precisely determine whether the key fob is close to the vehicle or not. Basically, the key fob is the prover and the vehicle is the verifier. The prover must send the aforementioned parameters, so that the verifier can measure distances and geographical position. Although the protocol looks secure enough, the attacker might be able to send the expected information through BLE, fool the verifier and still be able to perform a successful relay attack.

On the other hand, [JCL20] proposes radio frequency fingerprinting. They train a classification model with legitimate signals from different key fobs, so that malicious signals sent from a radio dongle or an SDR device can be detected. This mitigation is more robust, because car manufacturers can train their own models in order to define a baseline that contains all the radio frequency features of their devices, making it difficult for an attacker to bypass this anomaly-based countermeasure.

In [Gre+20], they use a timestamp so that, if the car receives an old signal, it discards the code and does not perform any action. This idea is valid as long as the time measurement is precise enough to prevent de-synchronization or other side effects.

2.3 Immobilizer

An immobilizer system is an antitheft device that protects the ability to start the engine. It requires cryptographic authentication from a Radio Frequency IDentification (RFID) transponder embedded within the key fob, so that it prevents an attacker from hot-wiring a car. In Europe, the immobilizer system has been mandatory since 1995 [Wou+20].

As in RKE and PKES systems, the majority of immobilizer cryptography protocols are proprietary. In [Wou+20], the researchers extracted the firmware from the immobilizer and reverse-engineered the cryptographic algorithm, which is called Digital Signature Transponder 80 (DST80), from Texas Instruments. They presented some examples where they could recover the cryptographic key and disable the transponder. Moreover, they found a way to recover the full 80-bit key using side-channel techniques.

Megamos Crypto is another protocol for immobilizer systems, based on Galois Linear Feedback Shift Register (LFSR) and Non-Linear Feedback Shift Register (NLFSR) schemes. It is analyzed in [VGE15]. Again, the researchers extracted and reverse-engineered an ECU firmware and found weaknesses in the cryptographic algorithm that led to the recovery of the 96-bit key.

In [Wou+19], a reverse-engineering process is shown to compromise the PKES system and the immobilizer in luxury vehicles. They show the use of an inadequate proprietary cipher with 40-bit keys and the lack of mutual authentication in the challenge-response protocol.

2.4 Cryptography

This section covers related work on cryptographic schemes used in the context of RKE/PKES systems and RFID immobilizers. Most of the research is about possible cryptographic attacks, side-channel attacks or cipher implementations such as Hitag2 and KeeLoq, which are widely used in the automotive sector.

2.4.1 Hitag2

In [GOKP16], the researchers did a deep analysis on the Hitag2 cryptographic algorithm, which is used by some manufacturers for rolling-code RKE systems as well as RFID immobilizer protocols. They successfully found a way to break the Hitag2 cipher with a few samples.

In [VGB12], the researchers found a way to compromise the cipher and unlock a vehicle in around 6 minutes with commercial hardware. Further analyses of the Hitag2 cryptosystem for RFID immobilizers and RKE systems can be found in [BRLK17], where an optimized exhaustive search algorithm is developed to forge Hitag2 radio signals from at least two eavesdropped rolling codes.

An OpenCL implementation to break Hitag2 is described in [Imm12]. Compared to FPGA implementations, it is not efficient, but it is less expensive. In

the worst case, they successfully break the cipher after 11 hours. An optimized guess-and-determine attack is proposed in [VVB18]. They claim that their implementation is capable of recovering the cryptographic key with 100% success rate and only two RKE signal samples.

2.4.2 KeeLoq

KeeLoq is a block cipher owned by Microchip that was used for rolling-code implementations. After the algorithm was leaked, a lot of research aroused to find vulnerabilities to break the cipher. A curated list of cryptanalysis and side-channel approaches on KeeLoq is presented in [End10]. Although there are ways to compromise the cipher, none of them seems to be applicable in RKE systems. Furthermore, KeeLoq is no more used nowadays, at least in rolling-code implementations.

KeeLoq is based on an NLFSR scheme, has a 64-bit key and operates on 32-bit blocks [Bog07]. Research has shown that KeeLoq can be compromised with slide attacks and meet-in-the-middle attacks, although the implementations require a considerable number of plaintexts and expensive hardware. [Ind+08] presents a practical key recovery attack against KeeLoq that requires 2^{16} known plaintexts. It is based on the slide attack and a novel meet-in-the-middle attack. The fully implemented attack requires 65 minutes to obtain the required data and 7.8 days of calculations on 64 CPU cores. A variant attack needs only 3.4 days on 64 CPU cores.

KeeLoq is also vulnerable to side-channel attacks. [Eis+08b] presents differential power analysis attacks in numerous commercial products that use KeeLoq as a rolling-code implementation. They successfully reveal the secret key of a remote transmitter and the manufacturer key stored in a receiver. As a result, a remote control can be cloned with only ten power traces, resulting in a practical key recovery in a few minutes. Once the manufacturer key is known, they demonstrate how to disclose the secret key of a remote control just by eavesdropping at most two messages from a remote key fob. Other researchers show that physical side-channel attacks can be used to recover the 64-bit KeeLoq key stored in a key fob [Eis+08a] or the 64-bit manufacturer key stored in the receiver [KKMP09].

2.4.3 Other ciphers

The researchers in [GOKP16] analyze the security of four vehicles of the Volkswagen group. The approach is gray-box-based since they were allowed to extract the firmware from the corresponding electronic control units (ECUs) to look for cryptographic objects and identify the cryptographic scheme. VW-1 vehicles are based on obscurity and an LFSR scheme, whereas the rest of the models use an 8-byte block cipher algorithm. VW-2 and VW-3 use a cipher known as AUT64, and VW-4 uses XTEA (eXtended Tiny Encryption Algorithm); both are Feistel block ciphers. The critical problem found is that all Volkswagen vehicles used the same global key for encryption. Therefore, once the key is found, any malicious user could clone a remote key fob to open any Volkswagen vehicle after eavesdrop-

ping a few signals. Despite being an important cybersecurity research, Volkswagen group did not allow the researchers to disclose all the information, methodology and techniques employed during the process. In [HGO18], these researchers continue to analyze the AUT64 block cipher, which is proprietary and closed-source. Again, they successfully identified some weaknesses that could compromise the algorithm.

2.5 Tire Pressure Monitoring System

The Tire Pressure Monitoring System (TPMS) has been investigated in [Rou+10]. These systems continuously measure air pressure inside all tires of a vehicle, and alert drivers if any tire is significantly underinflated. Although it is a safe-critical application, it can be misused in two ways:

- It can be used to track a certain vehicle, because it is an automatic protocol and difficult to deactivate.
- TPMS signals can be easily jammed or spoofed, because they use radio frequency communications. Thus, compromising the TPMS could lead to false dashboard warnings, among other consequences.

The TPMS protocol is very similar to RKE/PKES systems, because it uses the same frequency band (433 MHz in Europe and 315 MHz in the United States), modulation (2FSK and ASK/OOK) and encoding (Manchester). Furthermore, reverse engineering is needed to understand the protocol [Rou+10]. After that, it is possible to eavesdrop TPMS messages and even send arbitrary messages.

2.6 Global Positioning System

The Global Positioning System (GPS) is taking relevance in the context of V2V and V2X communications, since the entities involved in the dialogue must know the exact position of each other [PAS21].

As well as other wireless radio frequency technologies, GPS is vulnerable to jamming and spoofing, since GPS signals do not contain any information that can authenticate the source of the signals [PX21]. Several techniques for jamming GPS systems are presented in [FGSS18], using commercial SDR devices such as ADALM-Pluto, BladeRF, USRP or HackRF One [Gad24a] and open-source software such as GNU Radio [Rad24]. A curated list of attack scenarios to GPS as well as defenses and mitigations are shown in [PX21]. They describe the attack scenario, the criteria for defense strategies, and existing defense strategies; for both spoofing and jamming attacks of GPS signals.

Regarding detection mechanisms for GPS spoofing attacks, [SYZ22] shows several strategies, such as bias estimation range check, velocities consistency check, statistical test, least absolute shrinkage and selection operator, and global navigation satellite system augmentation.

2.7 Bluetooth

Bluetooth is another wireless communication protocol that is being used lately for several car functionalities. It uses the frequency band of 2.4 GHz and a distance range of up to 10 meters (although it can be extended to 100 meters). There have been several cryptographic designs on top of Bluetooth. However, due to inefficiency, most manufacturers avoid using these protocols [SYZ22].

One of the first Bluetooth cyberattacks was introduced in 2011 by Checkoway *et al.* [Che+11]. The attacker manipulated the vehicle radio system using a weakness in the Bluetooth stack due to the use of insecure functions such as `strcpy` in the C programming language [PX21]. Once the radio system was compromised, they gained access to the CAN network and sent messages to an ECU to disable the brakes.

In relation to the CAN bus, there are Bluetooth dongles that connect to the OBD-II port and communicate with a mobile application to show vehicle diagnostics and statistics to the end-user. It is interesting to analyze these types of functionalities to test if the Bluetooth connection is secure. In [WJL15], it is shown how a malicious mobile app can interact with the OBD-II connector and perform actions on the CAN bus. The researchers show a reverse-engineering methodology to analyze CAN messages in order to find the protocol structure; afterwards, the mobile application can get instructions from a back-end server and perform malicious actions on the car via CAN messages. They show a proof-of-concept video where the researchers control a vehicle remotely from their mobile phone (which is connected via Bluetooth with the OBD-II connector).

Bluetooth is also used for the keyless entry system of vehicles like Tesla. In [WGP21], the researchers show how they were able to exploit new attack vectors due to the use of Bluetooth. Particularly, the Tesla Model X uses Bluetooth Low Energy (BLE). The key fob exposed more attack vectors than traditional key fobs because of Over-The-Air firmware updates and a pairing functionality for registering new keys.

The uses of Bluetooth on car subsystems can have a huge impact. For instance, and adversary could gain access to the car if Bluetooth is used for keyless entry systems, and probably start the engine because modern vehicles implement PKES. On the other hand, an attacker could gain access to the CAN bus through an OBD-II connector.

In addition, car-sharing platforms rely on mobile applications that let users unlock their cars with a mobile application that connects via Bluetooth. If this communication is not secure enough, an adversary could find the flaws and exploit them to get unauthorized access to car-sharing vehicles. In [SMP16], the authors present a novel physical keyless car sharing system that allows car owners to generate digital keys for accessing their cars, and to share these keys with other users. It also provides a comprehensive analysis on the threats of car sharing systems.

2.8 OBD-II dongles

From the perspective of an adversary, one of the attack surfaces posing the highest potential for impact is the direct access to the vehicle's CAN bus, through which its various ECUs communicate between each other. The OBD-II system, whose inclusion and easy access for the purpose of vehicle maintenance is regulated in the European Union under the EU Regulation 2018/858 [Uni18], often allows to access the vehicle's CAN bus. While the implementation of an OBD-II port is mandatory under this regulation, research suggests an alarming lack of secure-design principles such as network segmentation [Kos+10].

As a result, access to OBD-II ports can be leveraged in order to exploit vulnerabilities found within these systems under a variety of attack scenarios and with a diverse spectrum of potential impacts. Access to the CAN bus of a vehicle through the OBD-II port may allow, for example, to interact with the car's windows in order to steal valuables inside [LH23], to unlock the vehicle by impersonating the ECU responsible for authenticating the electronic ignition key [Tin23], and even to disable the brakes while driving [Gre15; MV15; VM14; Che+11]. However, access to the OBD-II port tends to be limited to the interior of the vehicle, itself constituting a security boundary, thus limiting the usefulness of such an attack vector in most scenarios.

Outside its use in vehicle maintenance done by service centers, such diagnostic ports have demonstrated other popular uses, many of them largely materialized in the so-called OBD-II dongles: small devices that users can connect to the OBD-II port in order to gather information and telemetry from the vehicle. Figure 2.5 shows several examples: (1) BlueDriver (model LSB2) from Lemur Vehicle Monitors, (2) Bluetooth OBD-II Reader from Bafx Products and (3) generic ELM327-compatible device. Generally, interaction and data acquisition is done with the help of a mobile application which communicates with the device either through Bluetooth or Wi-Fi, allowing self-service access from the driver and/or owner of the vehicle, or through the use of a cellular modem (M2M, Machine-to-Machine) for use cases where centralized management is required. On the other hand, the direct access to the OBD-II port (and by extension to the CAN bus) that such devices rely on may potentially extend the attack surface towards the nearby proximity of the vehicle [LH23; GLP25b]; if the device uses cell connectivity instead, this may even allow for remote exploitation of such devices over the Internet [PM23].

2.8.1 Vehicle simulation testbeds

Some previous literature covers the development of simulation environments for vehicles that contemplate CAN-level ECU emulation, even going so far as to consider on-board diagnostics emulation. One such example is VEWE by Alvear *et al.* [ACCM15], a software-based end-to-end simulation which, unlike the work proposed in this article, emulates vehicle behavior which is then fed to the simulated ECU and emulates the OBD-II dongle itself, all in order to aid in developing and testing new devices that use OBD-II ports.

Figure 2.5. Examples of commercial OBD-II dongles.

Works that focus on vehicle cybersecurity can also be found, such as the recent VISE by Kabir and Ray [KR24], a digital-twin-based platform which is able to emulate a large number of ECUs, sensors and actuators for the purpose of simulating the complex effects of attacks within the CAN bus of vehicles. While our work in Chapter 4 requires the use of a hardware-based ECU in order to be able to physically plug in and interact with the devices under study, it is also able to use simpler off-the-shelf hardware that may only simulate an engine ECU. This is because our work is focused on assessing the security of OBD-II dongles themselves as a potential means to carry these CAN attacks wirelessly through an increased initial attack surface, and not CAN attacks themselves.

2.8.2 CAN-based attacks

We have previously provided an overview of various potential real-world impacts on vehicle safety and security that can result from access to the CAN bus by an attacker [LH23; Tin23; Gre15; MV15; VM14; Che+11; GLP25b]. While there are many attack scenarios that may be carried out within the CAN environment of a vehicle [Kos+10; HZB18], we focus mainly on attacks which can be carried out by sending arbitrary CAN frames through the OBD-II port. This includes spoofing and fuzzing of other ECUs exposed to the objective CAN bus, among others.

There exist several means to interact with the CAN bus from an offensive perspective which extend beyond the use of OBD-II ports. Which tool works best for a given situation will depend on how far the adversary is on the weaponization process (as described in Lockheed Martin's Cyber Kill Chain [HCA+11] cyberattack model). Early on in the development lifecycle, significant flexibility is required in order to quickly iterate and achieve results faster. Research may be thus better performed through off-the-shelf OBD-II dongles driven by *ad-hoc* code

running on the researcher's computer. This is already possible through software libraries and tools that allow researchers to interactively or programmatically interact with CAN buses through various means, such as `python-can` [Tho24]. As weaponization progresses, however, an effective application of the exploit may be done through purpose-built hardware devices [Tin23]. Table 2.1 summarizes selected open-source tools that can be used for automotive cybsersecurity research related to OBD-II and CAN.

The open-source body of software works also provides tools that focus on offensive exploitation of CAN buses. Some tools provide implementations of replay attacks ready for operational use, such as `CANalyse` [Lad21]; others assist the researcher with traffic analysis and reverse engineering, such as ATG [HZB18]. One of the most notable examples in this regard is Metasploit's own support for hardware devices, codenamed `HWBridge` [Rap24a; Smi17], which was originally contributed to the Metasploit Framework codebase, while not providing many advantages as of currently.

The proposed tool `pwnobd` (Chapter 4) provides increased flexibility in the development of device drivers and attacks by providing more granular common interfaces, of which a combination can be independently implemented by a given device driver, and similarly required by an attack implementation. This work also attempts to provide a more tailored and focused user experience that can adequately cover various operational stages as specifically needed for vehicle cybersecurity.

Table 2.1. Feature summary of CAN and OBD-II tools.

Tool	**Features**
`python-can` [Tho24]	Python library that provides common abstractions to different hardware devices, and a suite of utilities for sending and receiving messages on a CAN bus
`CANalyse` [Lad21]	Requires a hardware implant that sniffs, analyzes and uses the car information to command & control certain functions of a vehicle through a Telegram bot
ATG [HZB18]	Vehicle CAN bus packet analyzer and attack packets generator that enables researchers to explore automotive cyber-physical system rapidly, to analyze different attack modes in ECUs or to generate a dataset for later analysis
`HWBridge` [Rap24a]	General-purpose Metasploit module to connect to hardware devices such as CAN sniffers, among others. It is a functional interface to use the SocketCAN capabilities of Linux

2.9 V2X communication standards

Vehicle-to-Everything (V2X) communications are transforming modern transportation systems by enabling real-time information exchange between vehicles, infrastructure, pedestrians and networks. In particular, depending on the parts of the communication process, these concepts are known as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Infrastructure-to-Vehicle (I2V), Vehicle-to-Pedestrian (V2P) and Vehicle-to-Network (V2N). This integrated environment holds great potential for improving road safety, enhancing traffic flow and advancing autonomous driving technologies. However, the highly dynamic and decentralized nature of V2X systems also introduces critical security and privacy challenges.

The deployment of these innovative technologies requires the definition of appropriate standards; several Standard Defining Organizations (SDOs) have taken on this task. IEEE started the standardization work on 802.11p in 2004; the resulting standard was published in 2010. Further standardization work was initiated in 2006 by IEEE and ETSI, resulting in the IEEE 1609 series and the ETSI ITS series of standards being adopted in the US and Europe, respectively. In the US, the IEEE 1609 series of specifications led SAE to define application layer standards for V2V communication. IEEE 1609 is generally referred to as the Wireless Access in Vehicle Environment (WAVE), and SAE coined the term Dedicated Short Range Communication (DSRC). In Europe, the ETSI specification is referred to as Intelligent Transportation Systems (ITS-G5) or Cooperative ITS (C-ITS) [YP19].

Cooperative Awareness Messages (CAM) are defined in ETSI EN 302 637-2 [ETS19b], and Decentralized Environmental Notification Messages (DENM) are defined in ETSI EN 302 637-3 [ETS19a]. In the US, a similar concept is the Basic Safety Message (BSM), defined in SAE J2735 [Int22], which is analogous to CAM in ETSI ITS standards. Some message categories in BSM relate to DENM in ETSI ITS standards. CAMs are known as beacon messages, since they are meant to send information periodically [LM15]. There are other beacon messages such as Vulnerable Road User (VRU) Awareness Messages (VAM) defined in ETSI TS 102 894-2 [ETS20] and others. In contrast, DENMs are alert messages, since they are only sent when there is a special circumstance on the road.

Cooperative awareness in road traffic refers to the ability of road users and roadside infrastructure to be informed about each other's position, movement, and characteristics. Road users include all types of vehicles (cars, trucks, motorcycles, bicycles...), and even pedestrians. Road Side Units (RSU) can be elements like traffic signs, lights, barriers and gates.

This mutual awareness forms the foundation for numerous road safety and traffic efficiency applications, as outlined in ETSI TR 102 638 [ETS24]. This awareness is enabled through the regular exchange of information between vehicles (V2V, broadly including all road users) and between vehicles and roadside infrastructure (V2I and I2V), using wireless communication networks collectively known as Vehicle-to-Everything (V2X).

2.9.1 Cooperative Awareness Messages

The information exchanged to support cooperative awareness is encapsulated in CAMs, which are transmitted periodically. The creation, handling, and interpretation of these CAMs are managed by the Cooperative Awareness basic service (CA basic service), a component of the facilities layer in the ITS communication architecture defined in ETSI EN 302 665 [ETS10]. This service supports a variety of ITS applications, and is a required component for all types of ITS Stations (ITS-S), including vehicle ITS-S and personal ITS-S, that participate in road traffic.

CAMs are transmitted at a maximum rate of 10 messages per second in order to broadcast vehicle's information, which can be used to warn human drivers for dangerous situations or determine vehicles' maneuvers in autonomous vehicles [YP24]. The content of a CAM depends on the type of ITS-S. For vehicle ITS-Ss, the status information includes time, position, motion state, activated systems, etc. And the attribute information includes data about the dimensions, vehicle type and role in the road traffic, etc. When a CAM is received, the receiving ITS-S becomes aware of the presence, type, and current status of the sending ITS-S [ETS19b].

This information can support a variety of intelligent transport applications. For example, by comparing the sender's status with its own, the receiving ITS-S can evaluate the likelihood of a collision. If a risk is detected, it can alert the driver through the vehicle's dashboard or driver notification system [ETS19b].

2.9.2 Pseudonym identifications

Since CAM latency must be kept as low as possible to avoid impacting road safety, CAMs are sent in plaintext, eliminating the processing overhead of encryption and decryption. Another reason for not relying on encryption mechanisms is that sharing encryption keys over a dynamically changing topology where vehicles are constantly coming in and out of the communication range is not practical.

Therefore, malicious users can eavesdrop CAMs and obtain relevant information about vehicles [HS25]. For this reason, to preserve privacy on CAM transmission, messages hold a pseudonym ID instead of the real vehicle identity, as presented in ETSI TR 103 415 [ETS18b]. The pseudonym ID is supposed to be changed periodically, so that it becomes difficult to identify a vehicle based on such ID. The purpose of pseudonyms is not to anonymize messages, but to protect vehicle's privacy while not affecting traceability during a short period of time [YP24].

The use of pseudonyms protects privacy by making it more difficult to track a vehicle. However, the use of pseudonyms has several disadvantages. As stated in ETSI TR 103 415 [ETS18b], it is still possible to track vehicles regardless of pseudonyms, because the communication stack holds identifiable information in other layers of the communication (namely, the StationID in the Facility layer, the Geonet address in the Network & Transport layer, and the MAC address in the Access layer).

Moreover, pseudonym implementations have a trade-off between privacy and safety: the more frequently an ITS-S changes its pseudonym, the greater its privacy, since shorter pseudonym lifetimes make vehicles harder to track. Conversely, the less frequently ITS-Ss change their pseudonyms, the higher the accuracy of safety applications, as longer-lived pseudonyms reduce ambiguity and enable more accurate tracking of vehicle behavior.

When an ITS-S receives a new pseudonym, it is stored in a Local Dynamic Map (LDM) for a certain period of time. This can have consequences such as considering ghost vehicles when pseudonyms are no longer in use, missing vehicles during a pseudonym change process, or even the Sybil attack, which is possible in this context [ETS18b].

2.9.3 Digital signatures and certificates

Concerning the aspects of integrity, authentication and non-repudiation, V2X messages are digitally signed with digital certificates under a Public-Key Infrastructure (PKI) system. Due to the use of pseudonyms, the certificates used by vehicles are so-called pseudonym certificates. The specifications require the use of Elliptic Curve Digital Signature Algorithm (ECDSA), in which the public key length and the signature length are 32 and 64 bytes, respectively [YP23]. This is specified in ETSI TS 102 941 [ETS22], and the certificate formats are presented in ETSI TS 103 097 [ETS21].

Due to the fact that V2X communications must be very efficient to ensure road safety, the inclusion of pseudonym certificates in every message is not efficient in terms of speed and channel usage. One solution is to use Peer-to-Peer Certificate Distribution (P2PCD), where instead of sending the full certificate, the message contains the hash digest of the certificate. Receiving vehicles must hold a look-up table to find known certificates based on incoming digests. If no match is found, then the vehicles can request the full certificate and afterwards save it in the look-up table for later use [YP23].

Since a single vehicle can use several pseudonym IDs and thus several pseudonym certificates, they must request them to a Pseudonym Certification Authority. Instead of issuing one certificate at a time, which is not efficient, the Pseudonym Certification Authority will send several pseudonym certificates in a single message, namely using a butterfly scheme like the Security Credential Management System (SCMS) [Bre+18]. This scheme uses butterfly keys, which represent a novel cryptographic construction that allow a device to request an arbitrary number of public keys, each with a different private key and each encrypted with a different encryption key, using a request that contains only one verification public key seed and one encryption public key seed, and two expansion functions [Ham+17]. For instance, the IEEE 1609.2.1 standard uses a Butterfly Key Mechanism (BKM) protocol to efficiently request multiple certificates for use in V2X communication [BKS24]. In contrast, ETSI ITS mentions Issue First Activate Later (IFAL) [VHG19] in ETSI TR 103 415 [ETS18b] as a possible solution, although other schemes are proposed for standardization [Yao+21; DERL22].

2.10 Research gaps

Although there is already a considerable amount of research papers written on automotive cybersecurity, there are still topics that can be further investigated and topics that are emerging nowadays because of the use of modern technologies.

After a state-of-the-art review on automotive cybersecurity, these were the gaps identified:

- **Risk assessment**: The risks and impact of automobiles need to be measured. On the one hand, new regulations force manufacturers to pass cybersecurity tests [GHM23]. On the other hand, insurance companies must take this variable into account when defining the policy. Although risk assessment in the automotive sector is very complex, some approximations using well-known frameworks can be found in [Wan+21].

- **Supply chain**: The automotive industry employs a complex supply chain to source the components that are used to build new vehicles, provide services and perform repairs. This supply chain poses a huge risk to the industry, since each connected endpoint is a vulnerability waiting to happen [Dav23]. It is a fact that the security of individual components does not ensure the security of the whole system.

- **Reverse engineering**: Vehicle manufacturers will never publicly disclose any source code or detailed specification for the products they build. For this reason, researchers must utilize their reverse-engineering skills to extract ECUs firmware and analyze compiled artifacts from closed-source microprocessors, for example. Others might want to analyze radio signals or discover logic bugs in state-machine systems.

- **Cryptography**: On the one hand, there are protocols like RKE, PKES and TPMS that still lack robustness. Modern cryptographic schemes must be considered, such as lightweight cryptography [NIS22]. On the other hand, for devices that have no strict computational power limitation, post-quantum algorithms are preferred to prevent traditional algorithms to be potentially broken with quantum computers.

- **Mitigations**: While there is research about vulnerabilities and methods to compromise vehicular technologies, there must also be research about how to defend from these techniques and protect the affected systems.

- **Software security**: Modern cars are equipped with many features that improve the user experience and comfort. However, these new systems integrated in the car, such as operating systems or web browsers, must be analyzed. Manufacturers should follow best practices and continuous integration with quality assurance to prevent bugs and errors in their codebase.

- **V2X communications**: With the advances on technology, image processing, robotics and artificial intelligence, there has been a lot of studies on autonomous vehicles. These vehicles must communicate with each other (V2V) and with other entities (V2X). Standards are still being designed [5GA]. As a result, there is a need to protect these communications from the design phase, because they can be critical for human safety. Modern post-quantum cryptographic protocols are taken into consideration for this application, as shown in [YP23], [AK23] and [Lon+23].

- **Digital twins**: Digital twins are a cutting-edge approach to enhance vehicle security by creating virtual replicas of automotive systems and simulation environments [Hu+22]. In this context, security and privacy issues can be analyzed using a digital twin rather than a real vehicle [AFRR23]. This can help finding vulnerabilities and testing communication protocols [He+23], or even charging protocols for electric vehicles [BMR21].

- **Artificial Intelligence and Machine Learning**: AI can improve threat and anomaly detection, and intrusion prevention in connected and autonomous vehicles [LSWS25]. Machine Learning models can analyze vast amounts of real-time data to identify potential attacks before they cause any damage. Although training these models requires large, diverse datasets, AI can also generate synthetic data, which enables researchers to create simulation environments for a wide range of cyberattacks and rare threat scenarios [Li+23]. Simulations allow continuous testing and validation of security measures in a controlled environment. This approach accelerates the development of automotive cybersecurity and ensures safety in future intelligent transportation systems.

Chapter 3

Reverse-engineering and Breaking RKE Protocols

This chapter presents an in-depth analysis of the RKE system of many different vehicle models. Although there is a lot of research on this topic (namely, there are several types of keyless entry systems, mitigations, attack techniques and analysis approaches), this exhaustive analysis is not found in the literature. However, the main contribution is the development of the procedure to make the analysis, which proved successful with a vulnerability found during this research work. In brief, this vulnerability allows to clone a key fob and thus lock or unlock the doors of the car at any time, remaining unnoticed, with the only requirement of capturing a single signal from the target key fob beforehand.

The main contribution of this chapter is the definition of a methodology that is based on reverse engineering and employs commercial and open-source hardware and software, which is less expensive and time-consuming than other approaches documented in the literature while having a similar result. For instance, firmware extraction tools can be notably expensive and hard to adapt to different environments, apart from being invasive on the tested device. In addition, having access to the firmware is not enough, since embedded devices normally use machine architectures that might require an IDA Pro license to get support. In fact, there are manufacturers that prefer using custom machine instructions, which can be extremely difficult to reverse-engineer, as there might be few information available, and extra work must be done to be able to analyze the firmware [RG20]. In contrast, the methodology proposed here is based on a black-box perspective, which is easier to setup, easier to adapt to different vehicles, non-invasive and more accessible for academic researchers, while still being insightful for the analysis.

In the automotive sector, manufacturers tend to design proprietary cryptographic algorithms to generate short codes that improve the communication security. This fact makes vulnerability assessment much more complex, although still some insights can be obtained with the proposed reverse-engineering methodology. Furthermore, proprietary algorithms are less robust and more likely to be broken with cryptanalysis. Other manufacturers, such as Tesla, use secure cryptography and Bluetooth Low Energy (BLE) to implement RKE/PKES systems [WGP21].

The methodology and tools presented in this chapter were developed for evaluating the security level in RKE systems and discovering possible vulnerabilities through reverse engineering. For PKES systems, the level of security depends on the manufacturer implementation, including the protocols and the algorithms used. Reverse-engineering a protocol will not show a clear attack vector in most cases. However, this procedure will help determining and understanding the structure of an RKE protocol. If the protocol is weak, then it is prone to be compromised; otherwise, some insights can be drawn.

3.1 Radio frequency tools

This section contains some tools that can be used to audit radio frequency devices and perform cyberattacks. The list shows available commercial hardware tools and software tools, all of them are open-source.

3.1.1 Hardware

- HackRF One [Gad24a], from Great Scott Gadgets, is a half-duplex transceiver that operates at frequencies from 1 MHz to 6 GHz and can reach a rate of 20 million samples per second. With HackRF One and the appropriate software, the real-time frequency spectrum can be visualized. Moreover, it allows to capture and replay signals.

- There is another device from Great Scott Gadgets, known as YARD Stick One [Gad24b], that accepts different types of digital modulations (ASK/OOK, GFSK, 2FSK, 4FSK, MSK) and rates of up to 500 kbps. This gadget can be used to capture and decode signals, and to generate synthetic signals from binary information.

- Proxmark [Pro] is a tool designed mainly for RFID analysis and research. It allows testing, sniffing, replaying, and cloning devices such as RFID tags or Mifare Classic cards. Proxmark can be used to analyze immobilizers, which usually work as an RFID device. It can also be used to assess vehicles that have a PKES system to lock/unlock the car and even to start the engine, such as Tesla.

3.1.2 Software

- To interact with a HackRF One gadget, GNU Radio [Rad24] is a good option, which provides a graphical programming environment based on signal processing blocks for interacting with Software-Defined Radio (SDR) devices.

- GQRX [OZ924] is a program based on GNU Radio that displays the frequency spectrum in a waterfall model and is able to apply signal processing to received radio signals. It can be used to identify the working frequency of a given key fob with HackRF One.

- `inspectrum` [mie] is a software that displays the power of a signal in time and frequency. It is used to analyze signal capture files and extract their characteristics and even encoded symbols as bits.
- Universal Radio Hacker [PN18] is a project that encompasses the previous functionalities: it can be used to send and receive radio signals, but also to analyze their encoded information.
- `rfcat` [atl] is a Python library dedicated to the use of YARD Stick One.

3.2 Methodology

This section shows how the tools presented before are used in this work. Moreover, the reverse-engineering methodology is defined in order to obtain the necessary data to analyze different RKE protocols.

3.2.1 Signal captures

HackRF One is used to perform a basic reconnaissance of radio frequency and to capture key fob signals with GNU Radio. Figure 3.1 shows the process of capturing a signal emitted by the key fob and saving it to a file.

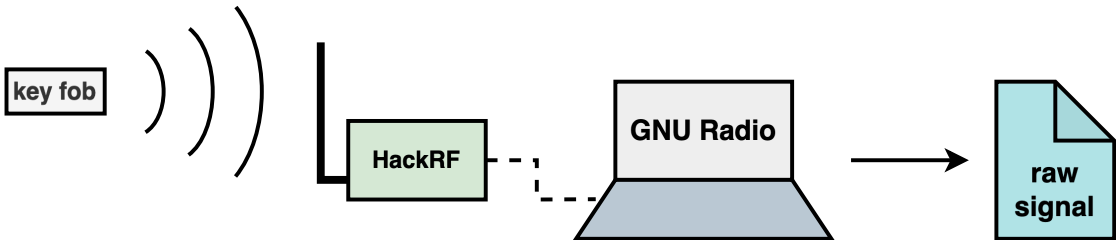


Figure 3.1. Signal capture procedure.

3.2.2 Signal analysis

The captured signals are processed by `inspectrum` and the power of the signal is plotted in time and frequency axes. RKE signals are usually modulated using digital schemes such as Amplitude Shift Keying, particularly On-Off Keying (ASK/OOK); or Frequency Shift Keying, especially with two carrier frequencies (2FSK). Figure 3.2 and Figure 3.3 show both kinds of modulations, respectively.

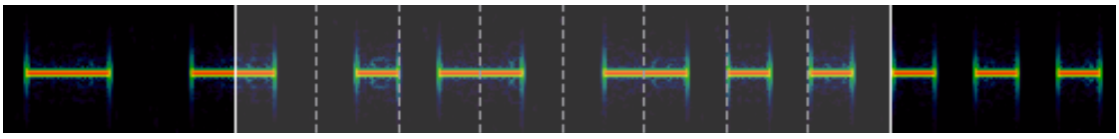


Figure 3.2. ASK/OOK-modulated signal.

Figure 3.3. 2FSK-modulated signal.

The signals carry modulated symbols, which can be extracted with `inspectrum`. It is common that the information bits are Manchester-encoded. In practice, this means that a bit `0` will be encoded as `01` and a bit `1` will be encoded as `10` (or the other way around, depending on the convention). Manchester encoding allows the receiver to keep synchronization at symbol level because there will not be a long period of continued signal power. Even a burst of `0`s or `1`s will result in an alternating sequence of `0`s and `1`s. As a result, `inspectrum`'s plot will show signal pulses with two different widths, so it is easy to identify that the signal is Manchester-encoded.

Furthermore, RKE signals usually start with a synchronization sequence. This is just a burst of `1`s. Figure 3.4 shows an example of a Manchester-encoded 2FSK-modulated synchronization sequence. The last bit of the synchronization sequence is `0`, which indicates the start of the information frame.

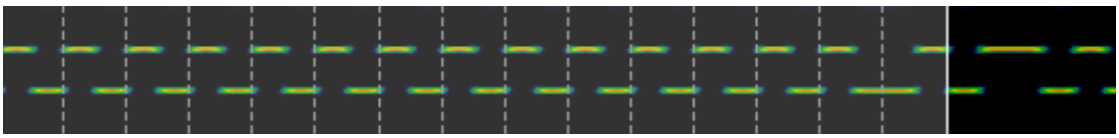


Figure 3.4. Synchronization sequence.

Once a signal binary data is extracted, several samples can be taken in order to look for coincidences and differences between distinct frames of the same RKE system. It is possible to guess several fields and extract features of the protocol. The overall process is depicted in Figure 3.5.

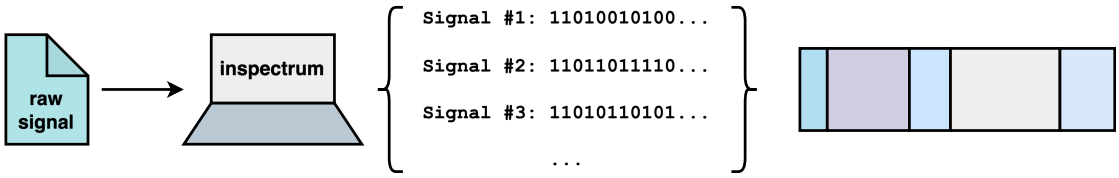


Figure 3.5. Signal analysis procedure to find protocol structure.

3.3 Results

In this section, several examples of RKE signals are shown and analyzed to guess the protocol structure. The relevant part of the encoded signals will be shown in tables, differentiating between commands ("Open" and "Close" and sometimes "Trunk" or "Lights"). Synchronization sequences will not appear in the tables.

Each signal has a number to keep an ordering (i.e. “Open 1” was captured before “Open 2”). Furthermore, the names of the tested models are not disclosed for privacy reasons. They are called Type A, Type B, etc. in the following sections.

3.3.1 Type A

This RKE key fob uses 2FSK modulation and encodes 168 bits with Manchester encoding. This is an example: `fffffffffffffe7b5fee66491d0592c722769755eb`. There is a synchronization sequence of 6 bytes `ff` ending with a byte `fe`. Therefore, the information bits are `7b5fee66491d0592c722769755eb`.

Table 3.1. Information bits for Type A.

Command	Code (hexadecimal)
Open 1	7bebee66491df2182cd18b4b59d4
Open 2	7ba3ee66491dc9fba0de9104d816
Open 3	7bfdee66491dca21ea6edadd9fc5
Open 4	7b0dee66491da5f7549b212f4660
Open 5*	7b3fedf0c83ccf4629f8133d4e07
Close 1	3b59ee66491d0585e74729e8e34c
Close 2	3b19ee66491df15688a309201984
Close 3	3b59ee66491d0585e74729e8e34c
Close 4	3b6cee66491d1f4902422d6b8f22
Close 5*	3b37edf0c83cc6b518a8185aae51
Trunk 1*	5d97edf0c83cb14d4a5ba82bee3f

The codes of Table 3.1 show some similarities. Code pieces that do not change between signals are highlighted in bold, whereas pieces that change between commands are underlined. As a result, the structure of the frame can be guessed. Figure 3.6 shows the information structure in bits:

- The first byte indicates the command (`7b` for “Open”, `3b` for “Close” and `5d` for “Trunk”).
- The second byte seems to be random, but it can also be a checksum for error detection.
- The next 4 bytes hold a key identifier. Signals marked with * have a different identifier because these signals were captured from another RKE key of the same car manufacturer.
- The last 8 bytes are random.

Figure 3.6. Protocol structure for Type A.

3.3.2 Type B

These signals are composed by two Manchester-encoded frames of 557 bits and 101 bits, using ASK/OOK modulation. This is an example of this RKE signal: `1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d1d0dd3492af800948c48` for the first frame and `1d1d1d1d0dd3492af800948c48` for the second frame. Both frames contain the same encoded information, which is the relevant part: `0dd3492af800948c48` (9 bytes).

Table 3.2 shows that only the first information byte seems to be predictable, the rest looks random. However, since there are no similarities between "Open" signals that differ from other commands, the signal bytes must be encoded or encrypted.

In other words, the vehicle must have a way to determine the command ("Open", "Close" or "Trunk"). Compared to Type A signals, there is no such byte that uniquely identifies the command. Therefore, the vehicle must decrypt the bytes to obtain the command from plaintext bytes.

Table 3.2. Information bits for Type B.

Command	Code (hexadecimal)
Open 1	`0dd3492af800948c48`
Open 2	`0cb762cd5c04b19d06`
Open 3	`0dba67ca8f08f9d0d4`
Open 4	`0c0492e9ec5c98035c`
Close 1	`0decc317c3c379a798`
Close 2	`0da277085711bbaa0c`
Close 3	`0dccc337ede37987b6`
Close 4	`0daf1c21db58a27b6a`
Trunk 1	`0d3ef6634fb4449dbe`
Trunk 2	`0c04e2e9ec5c9e170e`

Taking a closer look at the codes from Table 3.2, it can be seen that all codes start with bits `"0000110"` and end with a `0` bit. Therefore, it is likely that the relevant part is just 8 bytes rather than 9 bytes, since it probably comes from a block cipher.

3.3.3 Type C

This sends the same frame twice. It uses ASK/OOK and Manchester encoding. A single frame contains a 312-bit code: `fffffffffffffffffffffffffffffffffffffffffffffffffffeaf1cff2251a65b6563dfd8e3`. The first 27 bytes are always `ff` and `fe` at the end, forming a synchronization sequence. Hence, the relevant information is just: `af1cff2251a65b6563dfd8e3`.

The signals shown in Table 3.3 are very similar to the ones from Type A. Looking at the coincidences between the signals, the structure of the frame can be guessed:

Table 3.3. Information bits for Type C.

Command	Code (hexadecimal)
Open 1	af1cff2251a65b6563dfd8e3
Open 2	af1cff4baa59c9c16a0dabe3
Open 3	af1cff745dbba3e2abf38de3
Open 4	af1cffe43eda563cec4a7fe3
Close 1	af1cff1fa3d08921953959d5
Trunk 1	af1cffcf197cbf05a9861bb9

- The first 3 bytes are likely to be a key identifier.
- The next 8 bytes look random.
- The last byte indicates the command (`e3` for "Open", `d5` for "Close" and `b9` for "Trunk").

In fact, Type C is Volkswagen. The above frame structure is presumably correct because it is the same as the one disclosed in [GOKP16] (either VW-2, VW-3 or VW-4). Although the 8-byte field is not random but encrypted with AUT64 or XTEA, from a black-box approach it can only be stated that this field looks random or possibly encrypted. Figure 3.7 shows the structure of the protocol.

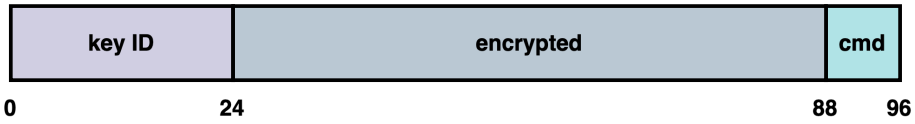


Figure 3.7. Protocol structure for Type C.

3.3.4 Type D

This key fob uses 2FSK modulation with Manchester encoding. One signal is composed of 4 frames. The first and third frames are just synchronization sequences, and the second and fourth contain almost the same information. Each information frame is composed of 172 bits and starts with 44 bits `1` (`fffffffffff` in hexadecimal). This is an example of such information frames (removing the initial 44 bits): firstly `93f1bdf759aa55953107544a0c69698f` and secondly `93f1bdf659aa55953107544a0c69691e`. Both signals are exactly the same except for a nibble that always changes from `7` to `6` and the last byte, which looks random.

After taking several samples, some coincidences were found. According to Table 3.4, this might be the structure of the information bits, depicted in Figure 3.8:

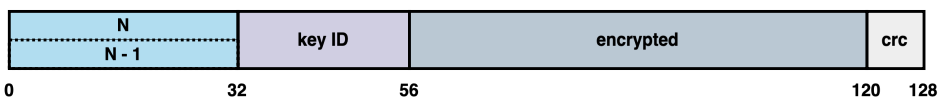


Figure 3.8. Protocol structure for Type D.

Table 3.4. Information bits for Type D.

Command	Code (hexadecimal)
Open 1	93f1bdf759aa55953107544a0c69698f 93f1bdf659aa55953106544a0c69691e
Open 2	93f1bdf759aa55b33e8293dce810aca2 93f1bdf659aa55b33e8293dce810ac33
Open 3	93f1bdf759aa55b250082f0c69e02913 93f1bdf659aa55b250082f0c69e02982
Open 4	93f1bdf759aa55edd8d1ae5697d527bd 93f1bdf659aa55edd8d1ae5697d5272c
Close 1	93f1bdf759aa5591feccdf4419205544 93f1bdf659aa5591feccdf44192055d5
Close 2	93f1bdf759aa55eba94c54636b8ca70f 93f1bdf659aa55eba94c54636b8ca79e
Close 3	93f1bdf759aa5525a98a4eedc41679b2 93f1bdf659aa5525a98a4eedc4167923
Close 4	93f1bdf759aa55f49c2593deeb3f7918 93f1bdf659aa55f49c2593deeb3f7989
Trunk 1	93f1bdf759aa55f96208816dbbfaf9f5 93f1bdf659aa55f96208816dbbfaf964
Trunk 2	93f1bdf759aa55df9aaf96090fea1cde 93f1bdf659aa55df9aaf96090fea1c4f
Trunk 3	93f1bdf759aa55548ed1268fac85b7e9 93f1bdf659aa55548ed1268fac85b778
Trunk 4	93f1bdf759aa55e1953c627b8c7e6549 93f1bdf659aa55e1953c627b8c7e65d8

- The first 4 bytes are always the same except for the last nibble, which is `7` for the first frame and `6` for the second.
- The following 3 bytes are always the same. This could be a key identifier.
- Then there are 8 bytes that look random between different samples.
- Finally, there is a byte that seems to be random, but it can be a checksum.

In fact, the last byte was really interesting, because between the two information frames of a given signal, this byte was the only one that was not predictable at first glance.

Using a tool called RevEng [Coo24], it was discovered that the last byte was, in fact, a Cyclic Redundancy Check (CRC) code of the previous part. The parameters of the CRC code could be extracted as follows:

```
# ./reveng -w 8 -s \
> 93f1bdf759aa55953107544a0c69698f \
> 93f1bdf659aa55953107544a0c69691e \
> 93f1bdf759aa55b33e8293dce810aca2 \
> 93f1bdf659aa55b33e8293dce810ac33 \
>
width=8  poly=0x2f  init=0xe0  refin=false  refout=false
xorout=0x00  check=0x56  residue=0x00  name=(none)
```

As can be seen, using only 4 codes (that is, 2 signals), the tool is able to obtain the CRC parameters. To verify that the result is correct, one can generate the CRC values of the following codes and see that they coincide with Table 3.4. Although this is a great outcome, it is not useful because the codes have an 8-byte field that looks random and does not seem to be predictable. Actually, it is likely to be encrypted because there is no clear command data, similar to Type B.

```
# ./reveng -c -w 8 -p 2f -i e0 \
> 93f1bdf759aa55b250082f0c69e029 \
> 93f1bdf659aa55b250082f0c69e029 \
> 93f1bdf759aa55edd8d1ae5697d527 \
> 93f1bdf659aa55edd8d1ae5697d527 \
>
13
82
bd
2c
```

3.3.5 Type E

These signals are modulated in ASK/OOK, and there are a total of 328 bits (encoded in Manchester) divided into three frames (132 + 132 + 64). The three frames hold the same binary data, although the last frame is truncated. Curiously, there is no synchronization sequence at the beginning. This is a full signal to unlock the car: `cfc4a7df931fe4af33f129f7e4c7f92bc cfc4a7df931fe4af33f129f7e4c7f92bc cfc4a7df931fe4af`.

As can be seen, the relevant information appears in one of the 132-bit frames, which is repeated for redundancy purposes: `cfc4a7df931fe4af33f129f7e4c7f92bc`. Table 3.5 shows some samples for the same type of vehicle:

Table 3.5. Information bits for Type E.

Command	Code (hexadecimal)
Open 1	cfc4a7df931fe4af33f129f7e4c7f92bc
Open 2	cf482fab7998e4e633d20beade6639398
Open 3	cf07599391cce40333c1d664e4733900c
Open 4	cf242fa179abe4cb33c90be85e6af932c
Close 1	cf5bac7f7e9de45433d6eb1fdfa77915 0
Close 2	cf03622c13d2e42933c0d88b04f4b90a4
Close 3	cf0e14c3e3c4e43f33c38530f8f1390fc
Trunk 1	cf8619c1a932e4ee33e186706a4cb93b8

Some bits are always the same in all samples (highlighted in bold). The underlined nibbles have two bits with the same value (namely `c-d-e-f`, `3-7-b-f` and `0-4-8-c`). The hexadecimal representation is not helpful for this RKE protocol.

These codes must be encrypted or encoded since there is no clear sign that indicates the action ("Open", "Close" and "Trunk"), as happened with Type B as

well. Nevertheless, these codes look more obfuscated, and the coincidences do not make sense. Therefore, it is hard to guess the structure of the frame or what those fixed bits are used for.

3.3.6 Type F

This RKE key fob uses ASK/OOK modulation and encodes 312 bits with Manchester encoding. This is an example: `ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffec010824347acc6906fe589af`. There is a synchronization sequence of 26 bytes `ff` ending with a byte `fe`. Hence, the information bits are `c010824347acc6906fe589af`. With one button press, the key fob sends three consecutive frames. This is the frame structure guessed from the codes of Table 3.6:

- The first two bytes are always the same.
- The third byte holds the command (the upper nibble: `8` for "Open", `4` for "Close", `2` for "Trunk"). The lower nibble is alternating between `2` and `0`.
- The last 9 bytes look random, but only 8 are random.

Indeed, 9 bytes is a bit odd to be the output of a PRNG or a ciphertext, since it is not a power of 2. As expected, RevEng shows that the last byte is a CRC code.

Figure 3.9. Protocol structure for Type F.

3.3.7 Type G

This manufacturer uses 2FSK modulation with Manchester encoding. Compared to other manufacturers, this one employs a complex protocol which is very difficult to reverse-engineer. Two different models of this manufacturer were analyzed (high-end and low-end cars). After analyzing some samples, these conclusions are drawn:

- A long synchronization sequence is sent.
- After a certain period of time there is a set of 15-byte separated frames. There are 3 codes in the first set for the low-end car and 4 codes in the high-end car.
- After some time, another set of frames is sent, which hold 9.5-byte codes and have a longer time separation. There can be from 2 to 9 codes in this set.

Table 3.6. Information bits for Type F.

Command	Code (hexadecimal)
Open 1	c010824347acc6906fe589af c01082 54e3bb57affb63f7c9 c01082fbd44797bfc2642195
Open 2	c01080d4573d6c61a184f93f c010808ac1ba67cd3683902e c01080ad5ae46ad65305fa53
Open 3	c01082918cf06d3f304828bd c0108274df5ae7203eb049a3 c0108 22322533c77f5e7c59c
Open 4	c010800d95fc58f9691730db c0108000687238b3e0ff528c c01080ff456d51d5afc56f06
Close 1	c010422a702fcc53ee1dfe75 c010421488d89e31c810c465 c010421a6410c991b69a5ad2
Close 2	c01040b09ba1412223a1b04b c010408 95c6bb5dc34da52fb c01040 57b7b795719fa49981
Close 3	c01042d271772a908cf86be3 c01042980e32d524ca54247d c01042912cb64b5cbcc75aaf
Close 4	c010404fd394e197a705e2ae c01040955db803b82ae7cd5b c01040c08d4c59935d7897e9
Trunk 1	c0102225859233948dfa015cb c010221 0d416fd0de3ba0089 c01022ad211492b03ed93a95
Trunk 2	c01020dff676f470988b8db5 c0102038 2651d6f98a9ea723 c010207af70dc9e3a0a93162

- Within a set, the codes follow a decreasing sequence with an offset that has a small variation. It is almost a linear decreasing sequence.
- In the first set, there are 7.5 constant bytes and then a hexadecimal digit that holds the command to be executed (`7` for "Open", `b` for "Close", `d` for "Trunk" and `e` for "Lights").

Table 3.7 shows some samples of this type of RKE signals. The synchronization sequence is omitted, and the two frame sets are separated by a blank line.

The frames of the first set look very similar among each other. They differ only in two bytes. The last byte seems to be unrelated, so removing the last byte and subtracting two consecutive codes result in a constant difference of `0x40000000000`. As in other RKE protocols, this situation is likely to have a CRC code on the last byte. Indeed, RevEng discovers a CRC setup for the last byte.

Curiously, the codes from the second sets were already used in other samples. Actually, the hexadecimal format for the second sets is not really helpful. These codes should be shifted 2 bits. For example, for "Open 1":

```
>>> hex(0x3fffb56007d3e57dfa8c << 2)
'0xfffed5801f4f95f7ea30'
>>> hex(0x3fffb56007d3e57df980 << 2)
'0xfffed5801f4f95f7e600'
>>> hex(0x3fffb56007d3e57df884 << 2)
'0xfffed5801f4f95f7e210'
>>> hex(0x3fffb56007d3e57df7b8 << 2)
'0xfffed5801f4f95f7dee0'
>>> hex(0x3fffb56007d3e57df6bc << 2)
'0xfffed5801f4f95f7daf0'
>>> hex(0x3fffb56007d3e57df5b0 << 2)
'0xfffed5801f4f95f7d6c0'
>>> hex(0x3fffb56007d3e57df4b4 << 2)
'0xfffed5801f4f95f7d2d0'
>>> hex(0x3fffb56007d3e57df3a8 << 2)
'0xfffed5801f4f95f7cea0'
>>> hex(0x3fffb4e007d3e57ff3a6 << 2)
'0xfffed3801f4f95ffce98'
```

As can be seen, the above shifts are similar to the signals from the first set of "Open 1". Except for the last result, all of them start with `fffed5801f4f95f7`, which is also similar to the prefix of the signals from the first sets. The remaining 2 bytes are not a CRC code according to RevEng. The part that might be changing randomly is a 5-byte chunk (underlined in Table 3.7), although it looks deterministic. Indeed, the sequence is decreasing, but the differences are not predictable:

```
>>> hex(0x676fb2ab23 - 0x664e133156)
'0x1219f79cd'
>>> hex(0x664e133156 - 0x656ebe537c)
'0x0df54ddda'
>>> hex(0x656ebe537c - 0x644f7a71f6)
'0x11f43e186'
>>> hex(0x644f7a71f6 - 0x636c7b18df)
'0x0e2ff5917'
>>> hex(0x636c7b18df - 0x62665b3aa3)
'0x1061fde3c'
>>> hex(0x62665b3aa3 - 0x61453a03fc)
'0x1212136a7'
```

The previous calculations were performed because the protocol seems to be related to timing. As an initial idea, it was likely that the key fob had some timer, and on each press, the current timestamp was sent as rolling code.

The reason behind this impression was a failed replay attack. A replay attack with HackRF One was attempted by capturing a signal from the key fob (far away from the car) and then replay the signal next to the car. On the low-end car, the attack did not work, but on the high-end car it was successful. The only difference in these tests was time: in the first attempt, the time elapsed between capture and replay was more than 10 minutes, whereas in the second attempt the whole attack was finished in less than 2 minutes.

Still much more effort needs to be put for this RKE protocol to fully reverse-engineer it. The fact that there are not many random bytes is promising, but the protocol is still obscure and complex to analyze from a black-box perspective.

Table 3.7. Information bits for Type G.

Command	Code (hexadecimal)
Open 1	fffed9801f4f95f7f8676fb2ab235e fffed9801f4f95f7f4676fb2ab2352 fffed9801f4f95f7f0676fb2ab2356 fffed9801f4f95f7ec676fb2ab234a 3fffb56007d3e57dfa8c 3fffb56007d3e57df980 3fffb56007d3e57df884 3fffb56007d3e57df7b8 3fffb56007d3e57df6bc 3fffb56007d3e57df5b0 3fffb56007d3e57df4b4 3fffb56007d3e57df3a8 3fffb4e007d3e57ff3a6
Open 2	fffed9801f4f95f7f8664e1331563 0 fffed9801f4f95f7f4664e1331563c fffed9801f4f95f7f0664e1331563 8 fffed9801f4f95f7ec664e1331562 4 3fffb56007d3e57dfa8c 3fffb4e007d3e57ffa82
Open 3	fffed9801f4f95f7f8656ebe537cf6 fffed9801f4f95f7f4656ebe537cfa fffed9801f4f95f7f0656ebe537cfe fffed9801f4f95f7ec656ebe537ce2 3fffb56007d3e57dfa8c 3fffb4e007d3e57ffa82
Close 1	fffed9801f4f95fbf8636c7b18dfd3 fffed9801f4f95fbf4636c7b18dfdf fffed9801f4f95fbf0636c7b18dfdb fffed9801f4f95fbec636c7b18df c7 3fffb56007d3e57efa80 3fffb4e007d3e57ffa82
Trunk 1	fffed9801f4f95fdf861453a03fc87 fffed9801f4f95fdf461453a03fc8b fffed9801f4f95fdf061453a03fc8f fffed9801f4f95fdec61453a03fc93 3fffb56007d3e57f7a86 3fffb4e007d3e57ffa82
Lights 1	fffed9801f4f95fef862665b3aa3a3 fffed9801f4f95fef462665b3aa3af fffed9801f4f95fef062665b3aa3ab fffed9801f4f95feec62665b3aa3b7 3fffb56007d3e57fba85 3fffb56007d3e57fb989 3fffb4e007d3e57ff98e

3.3.8 Type H

This type of vehicle was analyzed the same way as the rest, and it was detected that all the protocol fields are deterministic and predictable. The frame starts with a 40-byte synchronization sequence, and after that, the information is encoded in 8 bytes. Table 3.8 shows some examples.

Table 3.8. Information bits for Type H.

Command	Code (hexadecimal)
Open 1	b3bbe5b76c2f5d2c
Open 2	b3bbe4b76c2f5d8e
Open 3	b3bbe3b76c2f5d61
...	
Open 20	b3bbd2b76c2f5d55
Close 1	b3bbd1b76c2f5e4d
Close 2	b3bbd0b76c2f5eef
Close 3	b3bbcfb76c2f5e4b
...	
Close 10	b3bbc8b76c2f5ea4
Trunk 1	b3bbc7b76c2f57a1
Trunk 2	b3bbc6b76c2f5703
Trunk 3	b3bbc5b76c2f579a
...	
Trunk 10	b3bbbeb76c2f571b

As can be seen, the protocol structure is weak and lacks randomness and encryption. In other words, there is no unpredictable field, everything is deterministic and predictable. Figure 3.10 shows the structure of the guessed protocol:

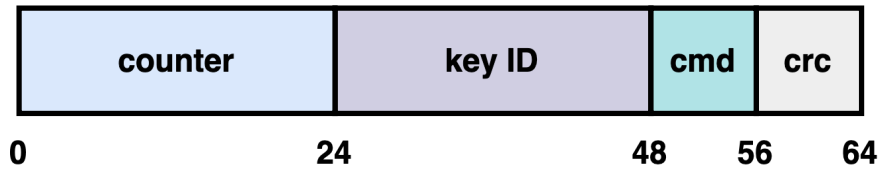


Figure 3.10. Protocol structure for Type H.

- `counter` is a 3-byte field that represents a decreasing counter, one by one.
- `key ID` is a fixed 3-byte number for a given key fob.
- `cmd` is 1 byte that encodes the action to execute ("Open", "Close", "Trunk").
- `crc` is an 8-bit CRC code to detect reception errors.

Before finding that the last byte of the frame was an 8-bit CRC code (like in the previous examples), a valid approach was to use a 1-byte brute-force attack, which is affordable. However, since the last byte can be computed as a function of the rest of the frame bytes, the brute-force approach is not actually needed.

As a result, if attackers wanted to compromise this vehicle, they only need a single signal capture before performing the attack. This would be the strategy:

1. Capture one signal.
2. Extract the key fob ID and the current counter.
3. Choose the command and decrease the counter.
4. Compute the CRC code.
5. Send the crafted signal.

After that, the signal will be valid and accepted by this vulnerable Type H vehicle.

This attack was implemented in a Python script using YARD Stick One and `rfcat`, available in GitHub [7Ro24]. Cloning the key fob was possible because the tool could send any key fob command.

A responsible vulnerability disclosure process with the manufacturer was started before submitting the corresponding paper [GLP25a]. They recognized the vulnerability and allowed the publication of our findings.

3.4 Discussion

Table 3.9 shows a summary of the previous analysis. For each type of vehicle analyzed, the modulation, number of information bits, number of random bits and the CRC parameters are listed.

Table 3.9. Summary of information bits.

Type	Modulation	Information bits	Random bits	CRC
A	2FSK	112	64	-
B	ASK/OOK	64	64	-
C	ASK/OOK	96	64	-
D	2FSK	128	64	`p:2f; i:e0`
E	ASK/OOK	132	-	-
F	ASK/OOK	96	64	`p:01; i:00`
G	2FSK	120	40	`p:01; i:fe`
H	2FSK	64	0	`p:7f; i:7a`

The “Information bits” field omits repetitions and synchronization sequences. Thus, it holds the bit length of the protocol frame. Analogously, “Random bits” shows the number of bits that are likely to be random or encrypted (in other words, not predictable).

All tested vehicles use ASK/OOK or 2FSK modulations, which means that only simple digital modulations are used to minimize the cost of the emitter and receiver modules on the key fob and the ECU, respectively.

It is interesting that "Random bits" field is 64 bits for 5 out of 8 models analyzed. This result is reasonable because of how block ciphers and PRNGs work. According to the number of unpredictable bits, the security of 5 out of 8 models is similar.

Notice how the protocol for Type B is completely random. This means that this manufacturer is using an encryption mechanism to grant security and privacy to the protocol. There is no way for an unauthorized eavesdropper to determine any protocol field (key ID, command, etc.) because they do not have the encryption key. Attackers must identify and cryptanalyze the cipher in order to break this protocol.

From an offensive point of view, none of the protocols (except Type H) is straightforward to compromise since the block cipher algorithms or PRNGs are proprietary and unknown. It can be said that Type G has a lower security level compared to the rest because it relies on a 40-bit value.

Nevertheless, using closed-source algorithms is security by obscurity, which is not recommended because these non-standard implementations usually contain weaknesses. Moreover, these algorithms tend to be leaked in underground forums, as happened with KeeLoq [End10]. Indeed, modern cryptography does not rely on private algorithms, but on private encryption keys.

Radio frequency attacks such as replay and relay attacks still work because of the way key fob communication is done. With this reverse-engineering methodology, researchers are able to learn how the information is structured in the messages sent from the key fob to the car, so that they can determine if there are vulnerabilities and the overall level of security of the car.

As a best solution, it would be more secure to use a standard block cipher such as AES, which is known to be very robust, although it would increase the hardware complexity of the key fob. However, if all vehicles used AES to encrypt their protocol frames, every captured signal would be indistinguishable from other models.

3.5 Conclusion

In summary, a reverse-engineering methodology to analyze RKE protocols has been presented. The main contribution is how a vulnerability was found in one of the tested protocols (Type H). It was exploited to predict next rolling codes with a single capture (in other words, clone a key fob). Once the protocol was fully reverse-engineered, YARD Stick One was used to generate and emit future signals, implemented in a Python script. In addition, other results have been presented using the proposed reverse-engineering methodology.

With this methodology, it is possible to quickly audit RKE protocols and determine if they are vulnerable or not, using a black-box approach, and commercial hardware and open-source software. Although having the knowledge and tools to extract firmware would make the reverse-engineering methodology more precise, it would also make the cost of the project much more expensive.

As a proof of the effectiveness of the proposed method, the results obtained for Type C (Volkswagen) protocol with reverse engineering were correct according to what other researchers did by analyzing the firmware of the ECU [GOKP16]. Nevertheless, the black-box approach has limitations. For instance, it is difficult to distinguish between random bytes and encrypted bytes. Plus, since manufacturers use proprietary ciphers, it is almost impossible to guess the algorithm from this perspective.

In order to improve RKE protocols, manufacturers need to implement a robust algorithm to encrypt the key fob messages. This way, none of the RKE protocols could be identified by the frame structure, and thus, the security and privacy would be preserved.

The automotive industry is in continuous growth and reverse-engineering methodologies need to be adopted by security researchers in order to evaluate the robustness of automotive protocols. This happens because vehicle manufacturers are not willing to publish or disclose any of their designs, which makes security analysis and vulnerability research much more challenging and difficult. On the other hand, new regulations have come into force and car manufacturers must pass a cybersecurity certification before selling their products.

Chapter 4

Penetration Testing for OBD-II and Bluetooth

In the field of vehicle cybersecurity research, one of the major challenges is the high cost associated with acquiring a physical vehicle to serve as a testbed for prototyping and demonstrating security attacks. As a result, simulation testbeds have become essential tools, offering cost-effective and accessible alternatives that allow researchers to evaluate and test the security of vehicular systems without the need for expensive real-world hardware. In addition, while it is possible to operationalize attacks in an *ad-hoc* manner, the resulting proof-of-concept programs may end constrained to a single vulnerable device model, further complicating scalable testing of the same vulnerability type against multiple devices. Thus, extensible software frameworks can help the researcher develop and launch device-agnostic attacks, progressing towards faster and more cost-effective operational implementations.

This chapter presents `pwnobd` [Nnu], a tool that comprises the aforementioned features. It was presented in Black Hat Europe 2024 (Arsenal) [GG], one of the flagship hacking conferences worldwide, which highlights its research interest within academia and industry regarding automotive cybersecurity. `pwnobd` is open-source, and installation and usage instructions, together with example code snippets, can be found in the associated GitHub repository [Nnu]. This chapter was developed with the help of Ignacio Gutiérrez.

The rest of the chapter is organized as follows: Section 4.1 further describes the identified requirements which in turn drive the design proposal. Section 4.2 describes the design proposal itself, contemplating the testbed platform itself and the `pwnobd` tool. Section 4.3 contextualizes the proposed design through a battery of tests performed on 3 different devices using the platform and a case study around CVE-2016-2354 [CMU16] vulnerability, describing how the proposed platform and the capabilities it offers could be utilized to identify similar vulnerabilities; the resulting findings are also discussed. Finally, Section 4.4 provides conclusions and outlines future research lines.

4.1 Requirement analysis

An OBD-II dongle communicates primarily with the CAN bus of the vehicle through its OBD-II port and interacts with other systems and devices through interfaces such as Bluetooth, Wi-Fi or cell connectivity (M2M) following the logic defined in its firmware. The primary objective of the `pwnobd` framework is to audit devices that communicate with an Android-based device running the appropriate mobile application, which in turn communicates with the device via Bluetooth.

In order to interact with the OBD-II device, its firmware exposes a high-level API in the form of a protocol overlaid on top of the corresponding wireless interface, through which the application can collect and perform the desired actions. Many devices implement the proprietary protocol offered by the ELM327 chip originally developed by ELM Electronics. As a result, it has become a *de-facto* standard, with a diverse ecosystem of devices and applications which support this protocol. Other devices implement proprietary protocols, along with specific mobile applications that interact with these devices.

In general, a significant number of these OBD-II devices communicate with the ECUs of the vehicle via the CAN bus protocol (or, on older vehicles, through other protocols like ISO 9141-2 [Int94]) in order to collect telemetry and/or diagnostic messages from the vehicle. This replicates a functionality similar to that offered by specialized tools employed in automobile repair shops for diagnosing failures within the vehicle's subsystems.

Derived from this context, various capabilities were identified. The platform design should provide these capabilities, focused on both the attack surface under study and the types of attacks to perform. The following attack surfaces are under scope:

- The mobile application that interacts with the device in a vendor-supported manner.
- The underlying communication layers (Bluetooth, Wi-Fi, etc.) through which the application seeks to communicate with the device.

Through the aforementioned attack surfaces, the platform should be able to perform the following attacks:

- Reverse-engineering.
- Low-level control, both of the CAN bus and of OBD-II dongles themselves.
- Man-in-the-middle interception of communications.
- Fuzzing of high-level protocols.
- Exploitation of firmware-level vulnerabilities.

Additionally, the following non-functional requirements are considered in the platform's design:

- **Accessibility**, providing ready-to-use capabilities and tools which provide a base allowing to perform attacks and collect results in less time.

- **Efficacy**, assisting security researchers in their audit and research workloads through the automation of simple attacks, allowing increased effort to be employed on more complex attacks.

- **Extensibility**, in order to support the continuous evolution of the platform as new technological innovations reach the commercial market.

- **Cost**, allowing a diverse audience to benefit from the capabilities offered by the platform.

4.2 Proposed design

As previously stated, the proposal has two components: a hardware testbed platform and the `pwnobd` software.

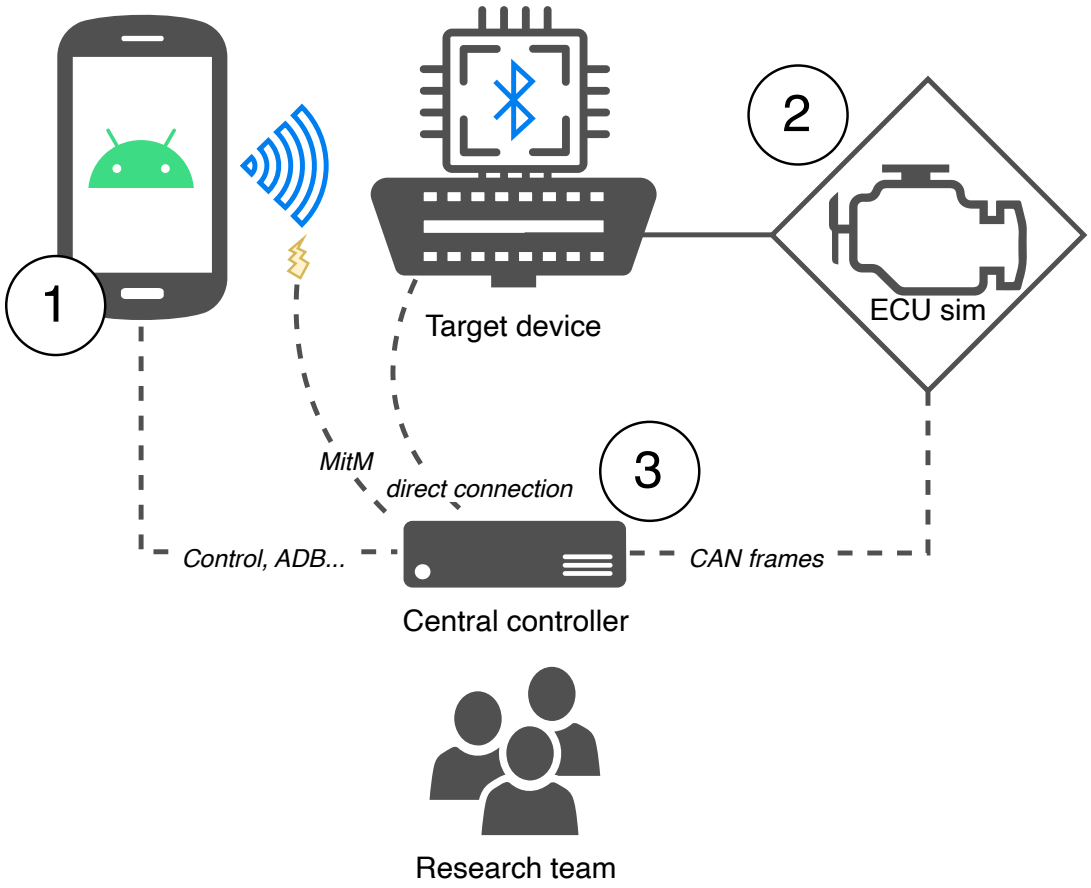


Figure 4.1. Design of the proposed platform with the primary elements.

4.2.1 Hardware platform

In order to address the identified requirements, a design is proposed with these components (see Figure 4.1 and Figure 4.2):

1. A central controller to centralize the workload and tools employed in the analysis and experimentation with the devices under test and the rest of the platform. The current version of this platform utilizes a Raspberry Pi 4B (with 8 GB of RAM), although it is possible to substitute this device with another of similar characteristics.

2. An ECU simulator, electronic device with an OBD-II port to which the target device is connected. The ECU simulator presents itself to the device as one of the vehicle's electronic control units, providing partial emulation of the messages that a vehicle could exchange through its communication buses. The current implementation of this platform utilizes a simulator device manufactured and distributed by Lianghung Co [Lia] for this purpose. This device allows for the simulation of multiple protocols in addition to the CAN standard, with an additional network tap capability over serial communication for capturing exchanged data over the communication bus in real time. It also provides easy access to the CAN bus itself through exposed through-hole pads within the circuit board, allowing for logic analysis and decoding using an oscilloscope.

3. An Android device on which the manufacturer-provided application for communicating with the target device is installed. This allows the researcher to test the target device as a part of the wider system it belongs to. The current implementation of this platform utilizes a Xiaomi Redmi 9.

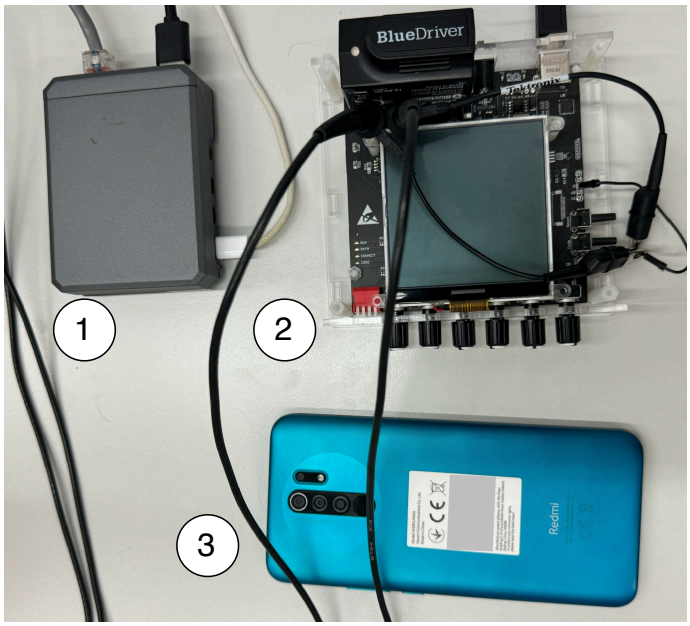


Figure 4.2. Implementation of the platform at the laboratory.

In regards to (3), the debugging and analysis capabilities offered by the Android operating system allow to study the interactions between the applications running on the device, intercept communications and firmware updates, and even modify the application's behavior through patches. This allows to experiment with a diverse repertoire of attack scenarios, from man-in-the-middle attacks on both the communication with the target and the communication with the Internet, to vulnerability exploitation performed on the OBD-II dongle or on the mobile application itself.

On the other hand, the use of an ECU simulator (2) simplifies the platform's design by averting the use of a real vehicle as a hard requirement, reducing the platform's total cost and allowing for riskier attacks that could compromise a real vehicle's availability and/or reliability.

The platform's various components are connected to the central controller (1), and the researchers interact with these devices through the tools available within it. For example, the central controller can connect to the Android device through USB or by using the Wi-Fi access point that the central controller itself provides, and interact with it through the ADB (Android Debug Bridge) protocol. Scripts are provided which expedite the connection setup process for wireless debugging, automatically detecting and connecting to suitable devices available on the network via multicast DNS. Simultaneously, the central controller can also monitor the ECU simulator's network tap and collect CAN communications from it, which can aid in attack verification and debugging.

Along with these capabilities, several tools are provided within the central controller to interact with the platform's components, to develop and perform attacks, and to collect results. One such tool is `pwnobd`, which will be further described in Section 4.2.2. Another tool that was provisioned within the central controller is MobSF (Mobile Security Framework) [Abr24], an open-source software package that provides static and dynamic analysis capabilities for Android applications. Through the use of scripts and automations to manage the system's components, the platform seeks to standardize security research procedures and improve efficiency, allowing the researcher to achieve results faster. In terms of performance, the proposed platform implementation is sufficient to meet the previously-specified requirements.

In addition, the central controller exposes several hardware interfaces, including Wi-Fi, Bluetooth, USB, SPI, I^2C and UART, among others. This allows integrating with additional hardware to arbitrarily expand and modify the platform's capabilities as the situation requires. For example, attacks on the Bluetooth communication layer could be performed either through the central controller's built-in Bluetooth adapter or through the use of external specialized devices. The central controller could also directly interact with the device under test through physical debugging interfaces such as SWD, JTAG or UART.

4.2.2 `pwnobd` framework

The hardware platform described previously can already provide significant value as a safe, cost-effective environment for security research on OBD-II devices by itself. However, it alone cannot fulfill all the stated requirements; for example, the platform may not provide increased efficacy as opposed to an *ad-hoc* testbed.

To fill this gap, we introduce `pwnobd`, a Python-based framework from which researchers can develop attacks directed towards OBD-II devices in a device-agnostic manner. The developed attacks are then exposed as part of a command-line tool from which an operator can scan and connect to supported devices and launch and manage attacks at scale. This enables the automation of simple attacks and provides assistance in the development of complex attacks.

In order to allow the researcher to develop device-agnostic attacks, `pwnobd`'s software architecture relies on an application of the bridge software design pattern [GHJV95], where the implementation of an attack does not need to rely on device-specific implementation details, but instead targets a framework-common set of interfaces, which can then be specified as required within the attack's definition (see Figure 4.3).

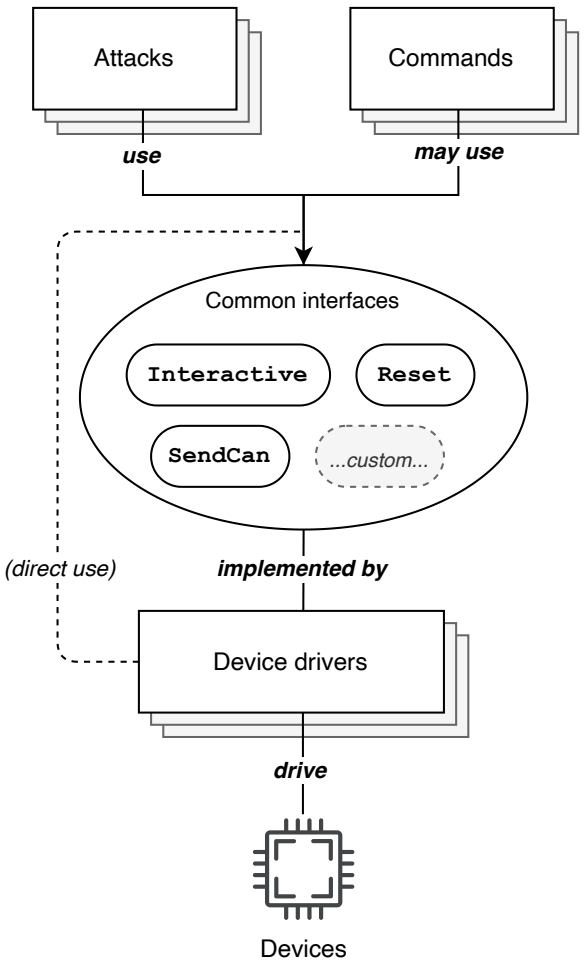


Figure 4.3. Architecture of the core abstractions provided by `pwnobd`.

An example of a common interface is `SendCan`, which provides functions for sending arbitrary CAN frames to the CAN bus through the device. In order to add support for a new device to `pwnobd`, a device driver is implemented. Device drivers allow to set up and teardown persistent connections to the device under study, providing the necessary logic to maintain the connection and interact with the device, along with implementations for any given common interfaces that it may support. By specifying the device driver's concrete class as a requirement, device-specific attacks (for example, fuzzing attacks against the device's high-level API) may be similarly implemented. Common interfaces may also provide functionality outside attack implementations; for example, implementing the `Interactive` interface allows an operator to open an interactive console from which they may send arbitrary commands to the device, and devices supporting the `Reset` interface can be returned to their initial state automatically after certain actions or manually through the use of a command.

An extensible system of scanners is also provided as part of the framework, which allows device driver implementations to provide additional logic to scan and identify nearby devices that may be compatible with such driver.

These elements constitute the software framework through which attacks targeting OBD-II devices can be developed. The user experience principles for this interface take inspiration from industry-standard penetration testing and red-teaming tools such as Metasploit [Rap24b] and `sliver` [Bis], seeking to provide the operator with effective tools which can be used to test the previous capabilities.

4.3 Use cases

In order to evaluate the suitability of the platform as a whole, two classes of evaluation work were performed:

1. A suite of general tests performed against various devices using the platform.
2. A more detailed scenario describing a follow-through of the research done as part of CVE-2016-2354 [CMU16].

4.3.1 General platform evaluation

In order to evaluate the general capabilities provided by the platform, several tests were conceived and carried out against the three OBD-II devices previously-pictured on Figure 2.5 (see Section 2.8). These include:

1. **Late connection test**, which seeks to identify whether the device accepts or reacts to connections done after an extended idle period of 10 minutes. This test seeks to model the device's susceptibility to be exploited in attack scenarios where the user of the OBD-II dongle forgets or does not wish to unplug the OBD-II dongle from the vehicle after its use, which would allow an attacker to approach the vehicle while unattended, connect to the plugged-in OBD-II dongle and perform subsequent actions on the target vehicle.

2. **Simultaneous connection test**, looking to ascertain whether multiple clients can simultaneously establish a connection and interact with the device at the same time. This would allow an attacker to interact with the vehicle through the dongle even if a legitimate device was already connected, which could impact the reliability of the device. Two different clients are used for this test: the Android phone, posing as the legitimate client, and the central controller itself, posing as the attacker. If the connection is successful, an attempt to inject messages to be sent to the vehicle is done in order to rule out additional security measures.

3. **Arbitrary CAN frame injection**, i.e. determining whether the device is able to send arbitrary CAN commands. Most OBD-II dongles, by default, do not send pure CAN frames on request, but instead wrap the data into an ISO 15765-2 message [Int24] which is then sent (often as multiple CAN frames, if necessary) using a fixed CAN arbitration ID. For the purposes of this study, an injection is considered successful if a way is found to send CAN frames of arbitrary content without ISO 15765-2 formatting [Int24] and with an arbitrary CAN arbitration ID. In order to verify that the injection is successful, the ECU simulator's sniffer is used first to determine successful injection of an arbitration ID, then the message content is verified on an oscilloscope. The ability to inject CAN frames this way can significantly increase the attacker's capabilities and potential impact once initial access to the OBD-II dongle is established.

4. **CAN fuzzing simulation**, in which a subset of CAN frames from the *Car Hacking: Attack & Defense Challenge 2020 Dataset* [Kan+21a; Kan+21b] were replied to the simulator using the device. This test is used as a simple exercise of the abstraction capabilities that `pwnobd`'s framework provides: the concrete, per-device CAN injection logic is provided within the implementation of the `SendCan` common interface for the corresponding device drivers; whereas the dataset loading and general replaying logic were implemented separately as an attack, targeting devices that implemented the aforementioned common interface.

Table 4.1. General testing results.

	(1)	**(2)**	**(3)**	**(4)**
BlueDriver LSB2	V	X	V	V
BAFX OBD-II Reader	V	X	V	V
Generic ELM327 device	V	X	V	V

The results of these tests, as performed on the current implementation of the platform, are available on Table 4.1 (firmware versions tested were 2.59 and 2.60), where V means vulnerable and X means not vulnerable.

All devices tested were not vulnerable to allowing multiple simultaneous connections, which they should block during the Bluetooth connection establishment phase. All devices seem to accept connections no matter how much time the device has spent idle. While BlueDriver devices were previously considered not vulnerable since CVE-2016-2354 [CMU16] was remediated [Blu18; WCL20], it was identified that both firmware versions 2.59 and 2.60 (the latest as of testing) of the BlueDriver LSB2 device no longer provide this protection, as the device was shown to allow connections and command execution even after the defined idle period of 10 minutes.

Finally, all devices were found to be able to send arbitrary CAN frames. While the procedure to do so for ELM327-based devices is known and documented by ELM Electronics [ELM18], the discovery of this capability within the newest firmware versions of BlueDriver LSB2 devices is particularly novel. The process of identifying this capability is described as part of the CVE-2016-2354 [CMU16] case study.

4.3.2 Platform case study: CVE-2016-2354

To contextualize the potential utility of this work, a research scenario is proposed using a real vulnerability (CVE-2016-2354) found in a commercial OBD-II device. Through this scenario, several applications of the proposed design were identified in which the platform and software can provide benefits for researchers.

On April 2016, research performed by Dan Klinedinst, one of the threat and vulnerability researchers employed by CERT/CC (Carnegie Mellon University), identified that the BlueDriver devices manufactured by Lemur Vehicle Monitors did not enforce authentication on the Bluetooth interface, from which the target vehicle's telemetry could be read using the manufacturer-provided mobile application [CMU16]. An adversary within proximity of a vehicle that was left with a BlueDriver device plugged in could potentially connect to the device via Bluetooth and send arbitrary commands to the target vehicle's CAN bus using the OBD-II port through which the device interacts with the vehicle. The OBD-II device thus becomes a wireless entry point through which additional exploitation can be performed against the car's subsystems. It should be underscored that this exploitation would not require physical access to the interior of the vehicle, and therefore could potentially be performed without access to the car key.

The discovery of this vulnerability received press coverage [Hon16; Rob16], and Lemur Vehicle Monitors quickly released updates for both its mobile application and its device firmware. The primary remediation measure, as stated by Lemur Vehicle Monitors support agents [Blu18] and later verified in published research [WCL20], caused the device to reject new connections after a period of 60 seconds, forcing users to physically disconnect and reconnect the device from the vehicle's OBD-II port in order to reset this timer. In addition, encryption was implemented on the communication protocol between the mobile application and the device, and the underlying protocol was changed to limit the device's own capabilities through a list of permitted actions that could be performed by the client.

As part of the research process of this vulnerability, the platform could provide benefits on various activities:

- The researcher can install the manufacturer's proprietary mobile application on the platform's Android device. Using the platform's provided static and dynamic analysis tools for Android applications, the researcher could identify the lack of authentication within the Bluetooth communication layer.
- Likewise, by analyzing the application's logic and intercepting Bluetooth communications between the application and the device, the researcher could determine that the device can be used to send arbitrary CAN frames through the device's OBD-II port, through which the vehicle's safety could ultimately be compromised.
- By reverse-engineering the communication protocol, modifying the application or even identifying a way to alter the device's firmware itself through a malicious firmware update, it would be possible to send arbitrary commands to the device. This would allow, among other attacks, the development of tailor-made fuzzing attacks against the device itself, for which IoT-specific techniques exist in academic literature [Che+18; Fen+21]. The success of these attacks could be determined by examining the communications performed from inside the CAN bus, which are monitored and relayed to the central controller by the ECU simulator.

In addition, the platform would allow to validate and evaluate the remediations performed by the manufacturer, verifying that the original vulnerabilities are mitigated and that additional vulnerabilities are not introduced. Through the offered capabilities, the platform can thus assist in various phases of the research process.

As part of the evaluation of the proposed platform, it was used to assess the stated remediations as applied in modern firmware versions of the BlueDriver OBD-II dongle (along its companion Android application).

- The device was set up on the platform's ECU simulator, the application was installed on the Android device and the system's behavior as intended was tested. Through this process, the device passively captures legitimate Bluetooth traffic through the use of Android's Bluetooth HCI (Host Controller Interface) snoop log debugging facilities (thereby averting the need to perform an over-the-air man-in-the-middle attack against the Bluetooth communication itself), which is later downloaded onto the central controller through ADB for further study. Initial traffic analysis revealed a stream of encrypted communications performed through Bluetooth GATT (Generic ATTribute Profile) characteristics.
- The application was analyzed using the automated tools provided by MobSF. The resulting decompiled source code and metadata provided useful information about the application's security posture. This includes the use of a software packer in order to protect the application's intellectual property.

No effort was put into breaking this protection, as the researchers were able to identify and reverse-engineer the cryptosystem responsible for encryption and decryption of in-transit traffic outside the protected code.

- Once the cryptosystem was reimplemented, traffic analysis revealed a command-based high-level API. Captured traffic logs were sufficient to implement a basic `pwnobd` device driver for this device which could initialize the device and allow the operator to interactively send and receive these commands. In order to protect the intellectual property of Lemur Vehicle Monitors, the released source code for `pwnobd` does not include any code relating to the BlueDriver device.

- Due to the significant amount of logic that the Android application had and the protection of relevant code and assets, further reverse-engineering of the application did not yield additional observations. Through traffic analysis, it was identified that all commands started with a common prefix, in a manner similar to the use of AT commands in the ELM327 protocol [ELM10]. In order to find additional commands, the researchers implemented a brute-force attack against the command-based high-level API using `pwnobd` framework. The attack relies on an error-based side channel to identify valid commands. Several additional commands were unearthed as a result.

- In order to identify the actions that the identified commands may result in, researchers utilized the interactive mode provided by the `pwnobd` software to quickly experiment with the changes caused to subsequently sent CAN frames. This included manually attempting to specify further arguments. Through this effort commands were quickly identified that allowed to arbitrarily modify the CAN arbitration ID and to disable ISO 15765-2 formatting.

Thus, it was identified that it is possible to send arbitrary CAN frames using a BlueDriver LSB2 device on firmware versions 2.59 and 2.60, thereby increasing the potential for ECU exploitation.

4.3.3 Limitations and potential improvements

During evaluation of the platform as was implemented, we identified several problems regarding the hardware procurement choices made, which limited the platform's potential capabilities. Future replications and other implementations of the platform described in this work may benefit from taking the following observations into consideration:

- We did not use a rooted Android device, which limited data collection and reverse-engineering capabilities. For example, Bluetooth HCI snoop logs could not be retrieved in real time. Instead, a time-consuming bug report request would have to be triggered through ADB, which would in turn contain the requested logs. Additionally, lack of rooting limited the researchers'

ability to perform dynamic analysis, which limited the scope of the research given the heavy protections used by some of the tested applications, such as code packing and obfuscation.

- The ECU simulator always assumes incoming packets use ISO 15765-2 format, an assumption which may not necessarily be true. As a result, while testing arbitrary CAN frame injection and sniffing traffic using the ECU simulator, the data section of frames collected was often corrupted because of CAN frames that did not include valid ISO 15765-2 format. Common examples of this corruption include data truncation and over-padding. In order to determine the root cause of this problem and independently verify the results observed in Section 4.3.1, the CAN bus itself was probed with an oscilloscope and we verified that the CAN frames data section had the full data content as requested.

4.4 Conclusion

This work has proposed, in concert with a set of functional and non-functional requirements, a design for a testing platform for cyberattacks directed to OBD-II dongles, along with the software `pwnobd`, seeking to provide cybersecurity researchers increased productivity in the process of auditing and analyzing vulnerabilities within these devices. Its utility was illustrated through the analysis and re-exploration of an industry-relevant case study, along with the evaluation of various devices within a basic test framework seeking to identify the target devices' susceptibility to become an entrypoint for further CAN-based vehicle exploitation. Furthermore, `pwnobd` [Nnu] was presented in Black Hat Europe 2024 (Arsenal) [GG], one of the flagship hacking conferences from all over the world.

As previously explained, a common application of OBD-II dongles is to provide users with additional vehicle telemetry and diagnostics, in a way similar to the canonical use of such a diagnostic port. This may often be done in order to aid repair and maintenance efforts by both professional mechanics and users who may want to perform vehicle self-repairs. These devices are also highly useful in various research fields involving vehicles, as they allow researchers to collect varied, precise and detailed real-time data from the vehicle itself. Even in consideration of the fact that these devices constitute a fairly effective and inexpensive solution for many applications, their potential for malicious and safety-compromising misuse underscores the need for user education to develop and maintain a safety-oriented mindset towards their deployment and use, as with any other tool that may be used to work on a vehicle.

While there exist effective security mitigations, such as adding connection timeouts or CAN frame allow-lists, that vendors may follow in order to reduce the risk of vehicular exploitation being performed through their devices, users should consider the risk that connected OBD-II devices may pose on vehicle assets as part of their threat modeling and take precautions in order to limit the potential for exploitation, such as unplugging the device while it is not actively being

used. Additionally, vehicle manufacturers should consider researching and deploying mitigations against the exploitation of ECUs vulnerabilities, particularly those that may impact the safety envelope, while balancing the potential use cases that OBD-II ports can provide. Some potential remediations for OBD-II may also not always be practical, such as limiting the content of messages that the device may be permitted to send through the CAN bus, as it may drive up the hardware requirements of such a device due to the need to store a database of allowed CAN frame contents to be sent per vehicle model within the device itself. Such mitigation must also be part of a secure firmware update mechanism as the only means to update this database, as otherwise it could be subverted by direct modification of either the database or the device firmware itself.

Future work may explore expanding the breadth of tests performed on a wider park of devices, potentially applying relevant methodologies such as IoTSF [IoT21] and BSAM [Tar24]. Additionally, further expansion of the platform may be performed, for which there is particular interest towards providing capabilities for Bluetooth connection interception and hijacking as an attack scenario. Additional software components and further integration between them within the platform could also be provisioned and experimented with, to increase the platform's capabilities. Finally, there is also potential for expanding the proposed architecture for the analysis of a wider variety of devices within the scope of the Internet of Things.

Chapter 5

Security Gaps in ETSI ITS Standards

In this chapter, we analyze ETSI ITS European standards (EN), technical specifications (TS) and technical reports (TR) in search for security issues that can potentially compromise road safety. For each of the issues found, we explain their impact and present scenarios where these issues can be leveraged by malicious users to cause cyber-physical damages. In addition, we simulate some of the scenarios to quantify the impact, and propose solutions to address these issues. In particular, we have focused on the use of CAM and DENM. We have also considered the use of pseudonym IDs, which are employed to preserve privacy in V2X contexts.

Regarding simulations, we use Eclipse SUMO (Simulation for Urban MObility) [Lop+18], which is a free and open-source traffic simulation suite. SUMO provides several tools to define roads, intersections, vehicles, etc. Moreover, it provides an external API integration with TraCI (Traffic Control Interface) using programming languages like Python, C++, Java or MATLAB. Among other tools that come with the SUMO suite, there is OpenStreetMap, which allows to select an area around the world in order to generate a realistic network for simulation.

The remainder of this chapter is organized as follows. Section 5.1 shows security issues found in the ETSI ITS standards. Section 5.2 presents several scenarios where the issues found can compromise road safety, and we quantify and analyze their potential impact. Section 5.3 describes related works found in the literature that address the issues and solutions found. Finally, Section 5.4 summarizes possible solutions to said issues and concludes the chapter.

5.1 Security issues

Having introduced the basic concepts of Cooperative Awareness Messages (CAM) and pseudonyms in Section 2.9, in this section we present the security issues found in the ETSI ITS standards. First, we define assumptions on security vulnerabilities, including adversaries' capabilities: adversaries can receive (eavesdrop) communication channels, store messages received from the communication channels, and inject messages to the communication channels. We also assume that adver-

saries are capable of modifying the On-Board Unit (OBU) that is responsible for handling cooperative awareness services and pseudonyms.

Based on our analysis of the ETSI ITS standards, specifications and reports, we conclude that the following types of attacks are possible.

5.1.1 Replay attacks

According to ETSI EN 302 637-2 [ETS19b] in Section B.3, there is a 16-bit CAM field called `generationDeltaTime` that holds the timestamp in milliseconds modulo 65536 (2^{16}). More specifically,

$$\texttt{generationDeltaTime} = \texttt{TimestampIts} \mod 65536$$

Where `TimestampIts` represents an integer value in milliseconds since `2004-01-01T00:00:00:000Z`. This field is defined in ETSI TS 102 894-2 [ETS18c] as a 42-bit integer, so it will reach the upper-limit around year 2143.

Regarding `generationDeltaTime`, a CAM m_0 sent at time t_0 milliseconds is equivalent to another CAM m_1 sent at time $t_1 = t_0 + 65536 \cdot k$ milliseconds for an integer k, due to the modulo operation. In other words, a CAM becomes equivalent to another CAM sent after a multiple of ~ 65 seconds. Consequently, an adversary could save valid CAMs and resend them exactly at the time of t_0 plus a multiple of ~ 65 seconds to keep its validity. There is no need to change anything in the message and hence the digital signature will still be valid as long as the vehicle keeps using the same pseudonym. The same issue happens with VAMs according to ETSI TS 103 300-3 [ETS23], Section B.1.3. However, this issue does not exist in DENM as it does not use the same `generationDeltaTime`.

The impact of this issue is very wide and depends on the attackers' creativity. For example, an adversary could save CAMs at critical timings (such as a sharp change of speed) and resend them in order to cause other vehicles to believe that this message reflects reality and thus provoke cyber-physical damages.

To our knowledge, there is no other literature that points out this issue. In addition, there is no clear reason for this field to be only 16 bits long (2 bytes), neither for the use of the timestamp modulo 65536 instead of the timestamp itself, other than saving space in the CAM and VAM message content.

If replay attacks were possible due to this `generationDeltaTime` configuration, a simple change on the ETSI EN 302 637-2 [ETS19b] standard would fix the issue: the `generationDeltaTime` field should have a larger size and the use of the modulo 65536 should be removed. For instance, using a 64-bit field (8 bytes), there would be a coverage of 2^{64} milliseconds; in other words, more than 500 million years, with only 6 additional bytes. Even with a 48-bit field (6 bytes), the maximum coverage would be around 8900 years. Only the use of 16-bit microprocessors could possibly limit this change in the `generationDeltaTime` field at the implementation stage, but it is not mentioned in the standards, and handling 64-bit integers should not be a problem for 16-bit microprocessors.

5.1.2 Identity-based attacks

Pseudonyms are meant to be changed periodically by each vehicle depending on fixed parameters (such as time, distance or number of V2X messages sent), or using other strategies, as presented in ETSI TR 103 415 [ETS18b].

The intrinsic problem comes with the fact that vehicles choose their own pseudonym, which allows for spoofing attacks and Sybil attacks.

On the one hand, in a spoofing attack, the attacker acts like a legitimate user to deceive the victim user into believing they are someone else. On the other hand, in a Sybil attack, an attacker creates multiple identities and then disguises as several fake stations in order to interfere with the normal operations of the system or profit from provided services [Ham+22].

Regarding solutions, it is difficult to find a novel solution that is precisely optimal. All solutions found in the literature (see Section 5.3.2) and even in ETSI TR 103 415 [ETS18b] have disadvantages and trade-offs between privacy and safety.

5.1.3 Grayhole attack

DENM messages contain types of information that need to be disseminated in wider area beyond a single vehicle's transmission range. This information includes warning situations such as emergency braking, a vehicle standing on the road side, or an approaching emergency vehicle. Therefore, vehicles that receive these DENM messages are expected to forward and propagate the messages by re-broadcasting it to other vehicles.

In a grayhole attack situation, one or more malicious vehicles can suppress forwarding DENM messages to prevent downstream vehicles from receiving these messages, thus denying these vehicles from taking necessary actions or preventive measures. The potential consequence is a dangerous situation for being unaware of the warning situations.

On the other hand, De-centralized Congestion Control (DCC) is a mechanism in which a vehicle self-regulates the channel access based on its monitoring of the channel usage, specified in ETSI TS 102 687 [ETS18a]. In other words, if a vehicle detects a channel usage is above the pre-determined threshold, it withholds its own transmission in order to prevent the channel from overflowing.

DCC is a necessary mechanism in medium access control (MAC) layer-level operations to maintain the channel usage within the limit. However, from the security perspective, it is equivalent to a self-inflicted grayhole attack situation, i.e. messages are intermittently dropped. Periodic messages such as CAMs are likely affected when this condition is triggered, compromising the effectiveness of what CAMs bring. Triggering this indirect attack is rather simple: one or more malicious entities can intentionally increase the channel usage by sending invalid bogus messages to trigger this DCC mechanism in honest vehicles. Furthermore, this issue is not easy to solve because it is intrinsic to DCC and similar MAC mechanisms.

5.2 Impact analysis

In this section, we present several scenarios where the previously-discussed security issues might compromise road safety.

5.2.1 Phantom vehicle attack

In this scenario, we assume that replay attacks are possible due to the ~ 65 seconds wrap-around in CAMs (see Section 5.1.1). As a result, an adversary is able to record a CAM at time t_0 milliseconds and replay the same message at $t_0 + k \cdot 65536$ milliseconds, for any positive integer k. Therefore, we can assume that the timing constraint is easy to handle, so we can focus on the replayed CAM itself.

We consider the term *phantom vehicle* to the concept of sending a CAM that states that there is a vehicle sending specific information at a certain position of the road, when there is no vehicle actually on that position. Hence, we assume that the attacker is able to send a valid CAM using specific hardware devices or by controlling an RSU.

5.2.1.1 Scenario A

We use a simple two-lane unidirectional highway. While a vehicle is driving through the highway, a phantom vehicle appears at a certain position of the highway in the same lane, as shown in Figure 5.1.

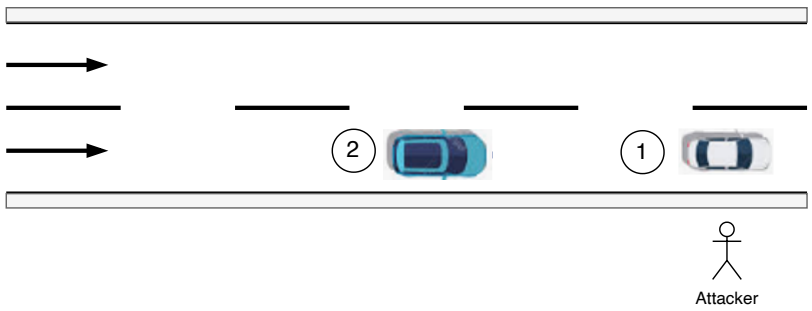


Figure 5.1. Scenario A, with phantom vehicle (1) and victim vehicle (2).

The issue is related to how the victim vehicle behaves with regards to the phantom vehicle. The ETSI ITS specifications do not mention any instructions to follow on this situation. We believe that the victim vehicle (2) receives the CAM from the phantom vehicle (1) and at least shows a warning message on the driver's dashboard. This might cause driver's reaction, or an automatic brake for safety reasons.

To build the simulation in SUMO, we have used a vehicle with default configuration to serve as the victim vehicle. This means that all parameters, such as speed or acceleration, have a default value assigned by SUMO, and the vehicle is fully controlled by SUMO. Then, we artificially insert a vehicle with TraCI to act as a phantom vehicle. We need to do this to let SUMO control the victim vehicle

and behave as if there is a real vehicle in front of the victim vehicle. After a small lapse of time, we artificially remove the phantom vehicle, so that SUMO increases the victim vehicle's speed back to a normal situation.

In the simulation, the victim vehicle drives from left to right, starting at $x = 0$ meters on the left. We place the phantom vehicle at time $t = 7.5$ and remove it at $t = 8.5$ seconds. We vary the position in the lane where it appears in order to measure changes in the speed of the victim vehicle. Figure 5.2 shows such speed changes depending on the position where the phantom vehicle appears. If the leftmost part of the road is $x = 0$ and the rightmost is $x = 200$ meters, we place the phantom vehicle in the interval from $x = 60$ to $x = 120$, in steps of 15 meters.

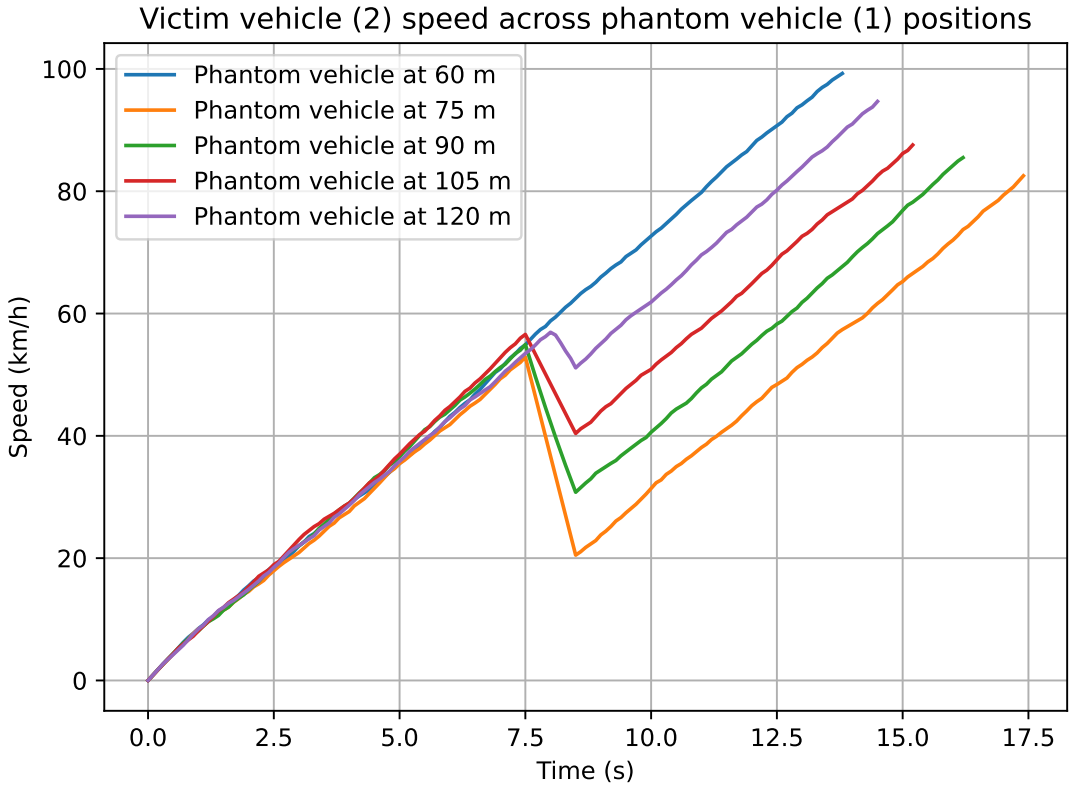


Figure 5.2. Results of the simulation for the scenario A.

As can be seen, if the phantom vehicle appears at position $x = 60$, the victim vehicle does not react because the phantom vehicle appeared behind it (failed attack). On the other hand, for the rest of phantom vehicle positions, the victim needs to brake in order to prevent a collision. Once the phantom vehicle disappears, the victim vehicle's behavior goes back to normal. Observe that the closer the phantom vehicle appears, the sharpest the change in speed is. Namely, for $x = 75$, (2) has to slam on the brakes to nearly stop the vehicle; whereas for $x = 120$, (2) brakes a little because the distance to (1) is bigger.

This scenario is a simple proof of concept that serves as a base for the following scenarios.

5.2.1.2 Scenario B

In this scenario, we start with the same configuration as in Section 5.2.1.1 but we add another vehicle. As a result, there are two vehicles driving through a highway, in the same direction. We assume that the leading vehicle supports V2X communication, while the other does not. This setup reflects a transitional phase during the adoption of V2X technologies, where legacy vehicles without such support still coexist on the road.

As depicted in Figure 5.3, when we insert the phantom vehicle (1), we assume that the V2X-enabled vehicle (2) will brake for safety reasons, as in scenario A. However, the legacy vehicle (3) must react manually to prevent a collision. Taking into account that (2) will brake suddenly and for no apparent reason (because the phantom vehicle does not physically exist), the reaction time from (3) can be long enough to cause a collision.

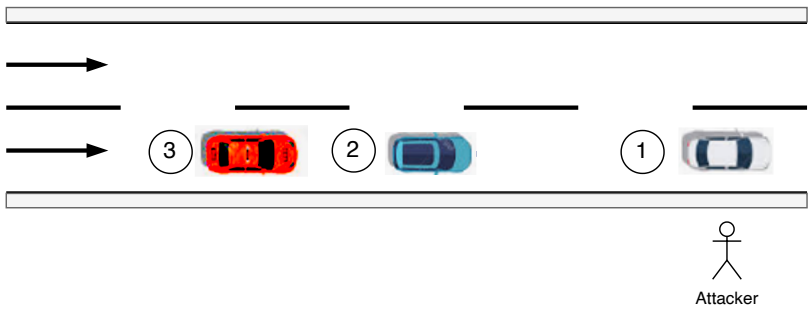
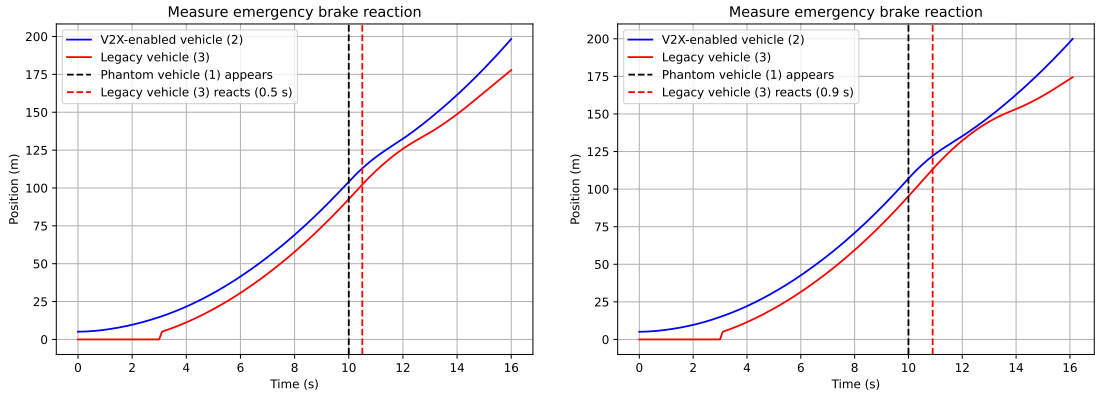


Figure 5.3. Scenario B, with phantom vehicle (1), V2X-enabled vehicle (2) and legacy vehicle (3).

We have simulated this scenario with SUMO and measured several reaction times for vehicle (3). In the simulation, the initial time-distance between (2) and (3) is 3 seconds (which corresponds to 100 meters distance at a speed of 120 kilometers per hour).

Figure 5.4a and Figure 5.4b show some of the results obtained from the simulation for different reaction times (0.5 seconds and 0.9 seconds, respectively). As can be seen, a reaction time of 0.5 seconds avoids the collision, whereas a later reaction time provokes a collision. Note that the blue and red graphs each show the position of the vehicles at a given moment, measured from their centers. Therefore, although the lines do not intersect, the vehicle centers are close enough to indicate a collision has occurred.

The impact of this scenario can be intensified if there are more vehicles in the road. In a real-world environment, this attack can cause a bigger accident affecting several vehicles. Furthermore, this scenario can have critical impact during the intermediate stage of V2X adoption, where older vehicles without connectivity continue to operate alongside modern and connected vehicles.



(a) Reaction time of 0.5 seconds.

(b) Reaction time of 0.9 seconds.

Figure 5.4. Results of the simulation for the scenario B.

5.2.1.3 Scenario C

In this scenario, we place several phantom vehicles at random positions within the road while there is normal traffic. We intend this scenario to be more realistic and measure the number of collisions caused by the inclusion of phantom vehicles.

Regarding the implementation, we have used OpenStreetMap to generate a SUMO network with several flows of vehicles, as Figure 5.5 shows. We chose the surroundings of the roundabout that crosses *Paseo de la Dirección* and *Calle del Capitán Blanco Argibay* in Madrid, 28029, Spain.

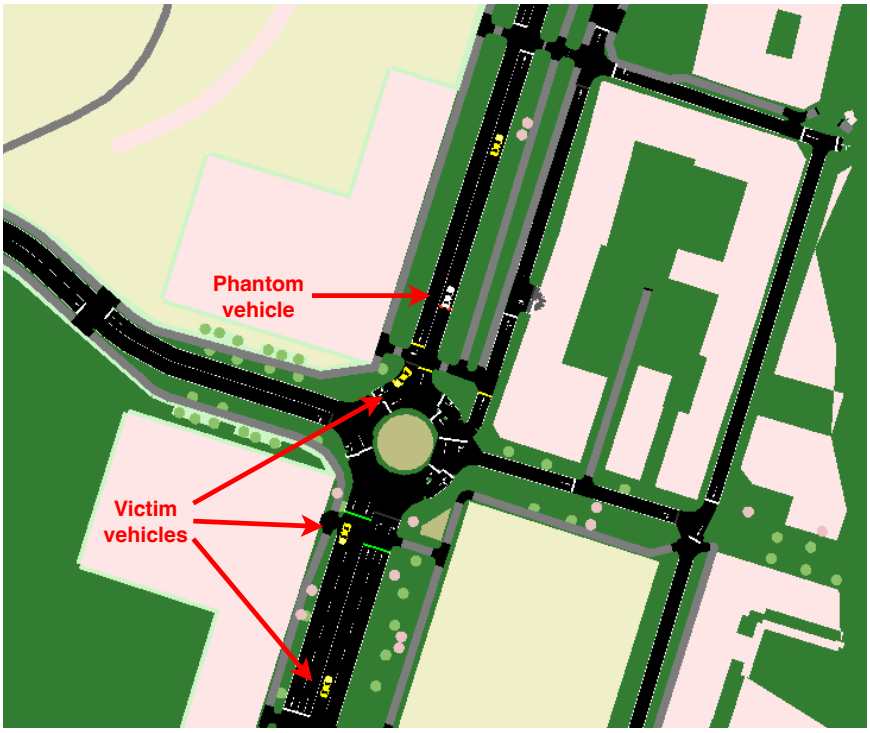


Figure 5.5. Scenario C, with victim vehicles (yellow) and phantom vehicles (white).

Here, we place 5 phantom vehicles, each of them at a random position of the road, at a random time; and we remove each of them at a random time as well. Figure 5.6 shows a histogram with the number of collisions measured in a total of 1000 simulation runs.

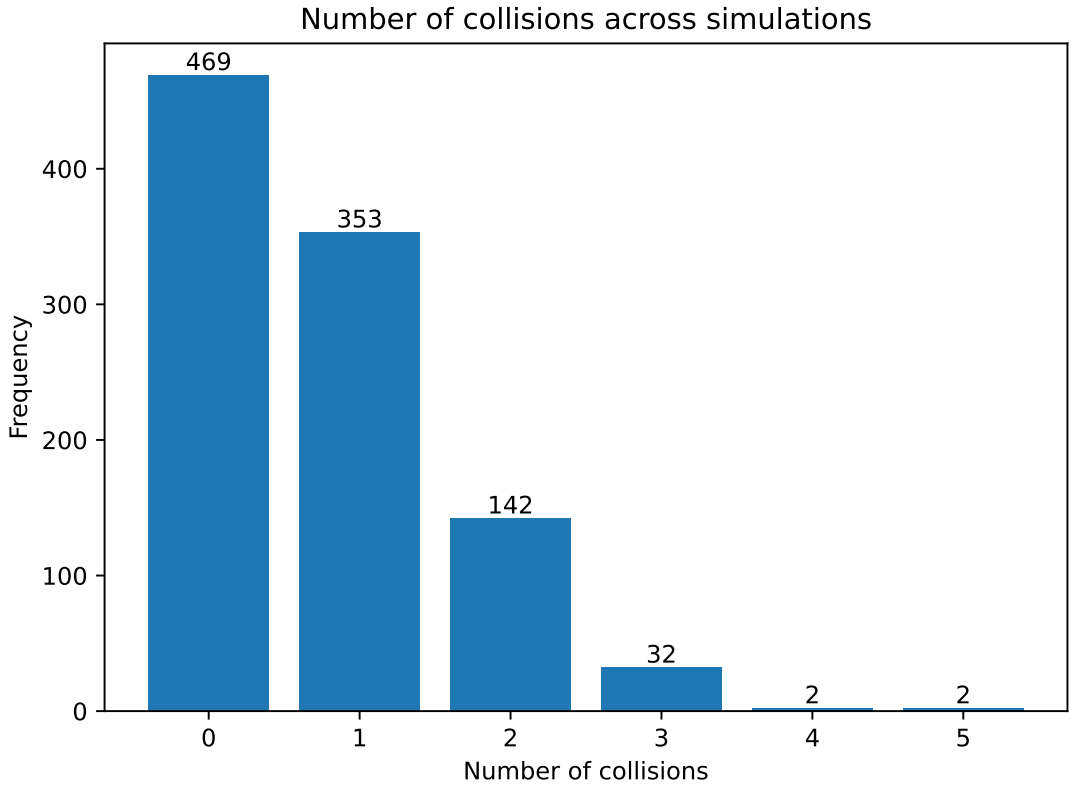


Figure 5.6. Number of collisions under a phantom vehicle attack.

As can be seen in Figure 5.6, more than 50% of the simulations show at least one collision, indicating a high likelihood that this attack can lead to hazardous scenarios where road safety could be compromised. In addition, the impact can be critical during a transitional phase where V2X-enabled vehicles coexist with legacy vehicles.

5.2.2 Identity-based attacks

As mentioned in Section 5.1.2, vehicles select their own pseudonym. Therefore, malicious vehicles can use this to perform spoofing attacks and Sybil attacks.

Figure 5.7 illustrates the concept of *pseudonym overlap* as an attack scenario, where the victim vehicle is changing naturally from ID-V-1 to ID-V-2, ID-V-3 and so on. When the attacker receives ID-V-2 from the victim vehicle in a message, they set the same ID from the victim to act like them, thus replacing ID-A-2 by ID-V-2. As a result, there is a time window where two vehicles hold the same pseudonym ID (ID-V-2). After a lapse of time, the victim vehicle changes to ID-V-3, and

the attacker performs the same actions to continue with the attack, so there is another time window where ID-V-3 is used by two distinct vehicles. This situation can continue indefinitely, and there is no easy way of determining which vehicle holds a legitimately generated ID and which one holds a stolen ID. Consequently, attackers may cause confusion on honest vehicles trying to determine which vehicle is which.

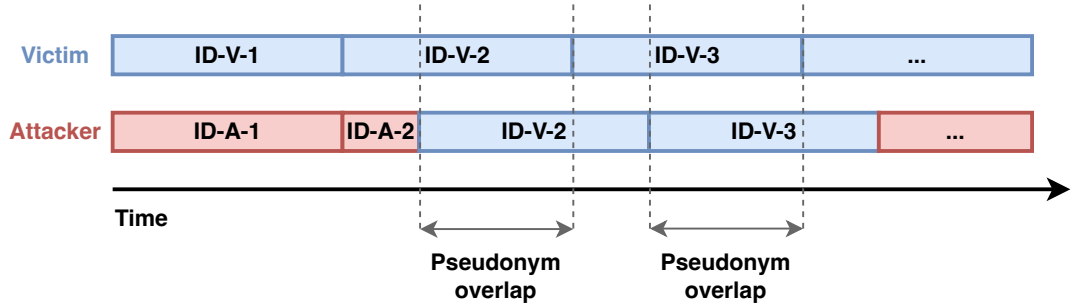


Figure 5.7. Pseudonym overlap represented in time domain.

We did not include simulation scenarios for identity-based attacks because, in the absence of reliable data on how vehicles would behave in such situations, there is currently no practical way to design realistic and representative experiments.

5.2.2.1 Spoofing attack

Figure 5.8 shows an example of this attack. The malicious vehicle (1) takes the pseudonym of vehicle (2) and uses that to send messages to the victim vehicle (3), while vehicle (2) is sending honest messages at the same time.

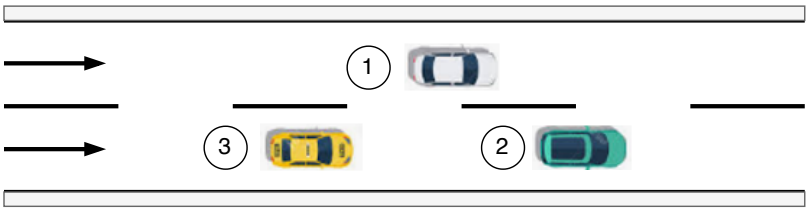


Figure 5.8. Spoofing attack scenario.

The victim vehicle (3) is confused because it is receiving mixed information from two vehicles holding the same pseudonym. This might result in assuming that both messages come from the same vehicle. For instance, vehicle (1) could tell that vehicle (2) is performing an emergency brake (when it is not), so that (3) brakes hard and causes collision with vehicles behind it.

5.2.2.2 Sybil attack

In a Sybil attack situation, the attacker vehicle is able to generate fake pseudonym IDs. The result is that victim vehicles believe there are several vehicles in the surroundings due to the use of different pseudonyms.

Hence, one possible scenario is the one depicted in Figure 5.9, where (1) is the attacker vehicle, (2) is the victim vehicle and the blue vehicles are generated by (1) as a result of the Sybil attack. This scenario may be viewed as the large-scale phantom vehicle attack, described in Section 5.2.1, where the attacker vehicle sends fake CAMs with varying position data to generate phantom vehicles.

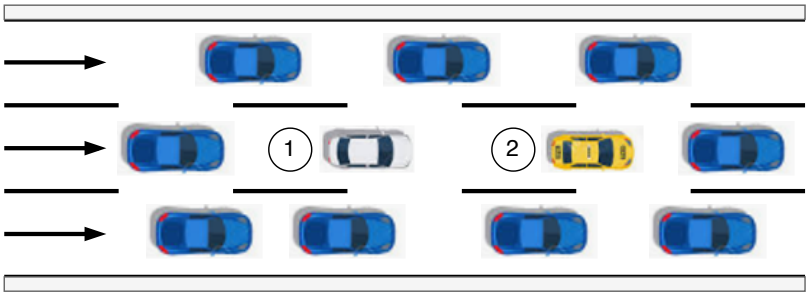


Figure 5.9. Sybil attack scenario.

Although the blue vehicles in Figure 5.9 do not physically exist, the use of this pseudonym implementation forces vehicle (2) into believing those blue vehicles are real. Moreover, there is no way to tell if a pseudonym ID is real or fake, because they are supposed to be generated at random and not traceable.

This attack can be used to cause a fake congestion. As a result, the road condition monitoring and decision-making of the CA services can be easily confused and misled, causing Denial of Service (DoS) attacks and even traffic accidents. Once real vehicles trust Sybil vehicles, malicious vehicles can successfully mislead real vehicles [Zhu+24]. Moreover, it can cause autonomous vehicles to slow-down or even stop in the middle of the road because of the fake congestion.

5.2.3 Grayhole attack

As explained in Section 5.1.3, a grayhole attack results in messages that are sent and never reach the target receivers. A similar situation can occur due to the DCC mechanism, since it withholds transmission, thus withheld messages are never sent and never reach the target receivers.

Under these circumstances, victim vehicles might be sending messages, but they rarely receive messages from other honest vehicles. This leads to a situation equivalent to not having V2X capabilities, which might come suddenly and affect driving behavior. In the case of DENM, it also leads to a loss of V2X capabilities, so victim vehicles must handle environmental issues manually. In this situation, unsent messages cannot be detected by other vehicles; in other words, things that never happened cannot be detected.

Since we lack reliable information about how vehicles behave in this situation to implement it with SUMO, we considered this scenario only as a theoretical case.

In fact, a possible scenario for this attack is similar to Scenario B (Section 5.2.1.2), but under slightly different assumptions. For instance, (1) could be a real vehicle that is stopped in the road, (2) receives a DENM warning about

this situation but does not broadcast this information (grayhole attack), so that (3) is unaware and might be in danger, as shown in Figure 5.10.

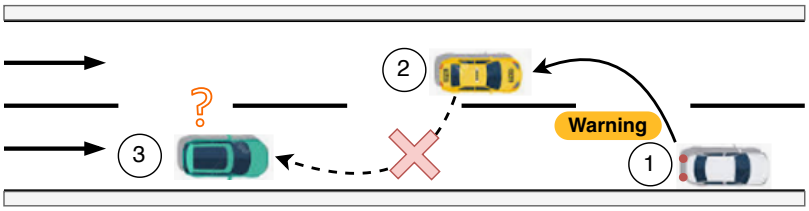


Figure 5.10. Grayhole attack scenario.

5.3 Potential solutions

After finding the security gaps and issues presented in Section 5.1, we conducted a literature review to determine if these are already covered in other research works.

A comprehensive analysis of V2X-related security issues is presented in [El-+20], along with other challenges regarding automotive cybersecurity.

5.3.1 Replay attacks

To our knowledge, there are a few research papers that mention replay attacks on the context of vehicle communications and try to find countermeasures.

A solution based on self-validating nonces is proposed in [LN24]. The authors claim that traditional countermeasures, such as timestamps and sequence numbers, have limitations in V2X environments because of intermittent connectivity and time synchronization. Self-validating nonces are cryptographic values that inherently validate their own freshness and uniqueness, eliminating the need for external verification mechanisms. They claim that once a self-validating nonce has been used and validated, it cannot be reused or replayed without being immediately detected as invalid. However, the self-validating nonce seems to be only a random number concatenated to its hash digest. It is unclear how vehicles can detect replayed messages based on invalid nonces. Furthermore, the paper does not consider a nonce-reuse attack.

An anti-replay attack mechanism based on hashing chain is proposed in [HNBS25]. This means that each message holds the hash digest of the previous message. The authors state that vehicles are responsible for generating and sending communication messages with hash chain information, RSUs verify and forward messages, and the cloud server is responsible for global management and monitoring. Therefore, since each message carries constantly updated hash chain information, a simple replay attack of previously captured messages will cause the hash value verification to fail. Nevertheless, this solution increases the complexity of the system due to the existence of concurrent messages and the use of another party (cloud server) within the communication system.

A survey paper [ASJ19] says that replay attacks can be prevented in LTE-V2X because of a timestamp and a nonce in each message. This claim is referenced

from [SNMB10], but there is no clear information about this fact. Another replay attack situation related to VANET routing is presented in [Fan+16], but it is not related with the kind of replay attacks under consideration in the current chapter.

5.3.2 Identity-based attacks

There are several approaches to protect pseudonym-based systems where privacy must be preserved but anonymity is not desired.

A comprehensive survey on pseudonym mechanisms in vehicular networks is presented in [PSFK15], where the authors analyze existing approaches based on public key, identity-based cryptography, group signatures, and symmetric authentication, while also outlining standardization efforts and open research challenges. A pseudonym scheme based on lightweight cryptographic mechanisms such as hash functions, quadratic residues and Legendre symbols is proposed in [NHZG25], and it ensures secure authentication, group key sharing and pseudonym management.

Regarding pseudonym overlap, [Tao+24] presents a protocol based on a Zero-Knowledge Proof (ZKP) protocol to prove that a given vehicle does not own a pseudonym. The authors limit their solution to a small environment, so that they can detect a local Sybil attack. They accept that global Sybil attack detection is hard to achieve. However, the benefit of this ZKP protocol is still vague.

Focusing on the locality assumption, the ring signature scheme proposed by [Mun+20] can be useful, which inherently preserves anonymity on the ring members. The main problem is that ring signatures require sending more data through the network. For instance, one signature on a ring of n members is a list of n public keys, n signatures and an additional ring signature. In the context of ECDSA (public keys are 32 bytes and signatures are 64 bytes), it will result in $32n + 64n + 64 = 96n + 64$ bytes. Therefore, it increases significantly the message size, and that is not desired. In addition, the P2PCD mechanism will be difficult to implement and PQC algorithms will make a ring signature scheme infeasible for V2X messages due to their large signature and public key sizes [YP23].

Other ways to detect Sybil attacks are presented in [NSSP04; XYG06], which rely on radio frequency measurements such as channel usage and signal strength. Other methods employ a timestamp series approach where the nodes attach a timestamp received from a recently crossed RSU [PAZ13]. Moreover, it is also possible to analyze the trajectories of adjacent nodes to detect Sybil attacks, presented as RobSAD (Robust method of Sybil Attack Detection) in [CWHZ09]. A deterministic Sybil attack detection approach is presented in [TB19], where the authors propose the inclusion of a field in the pseudonym certificate to forbid the overlap of a recently-used pseudonym.

5.3.3 Grayhole attack

There are several approaches to detect grayhole attacks, and can be classified as: infrastructure-based, where RSUs are used to detect misbehaving vehicles; and infrastructure-less, where vehicles nodes in the network identify misbehaving

nodes [Wan+20]. Among the techniques used to detect grayhole attacks, [Wan+20] describes reputation-based detection methods, ACK-based detection methods and other detection methods. Prevention methods consist of announcing malicious pseudonym IDs to the network, so that routing protocols avoid them and malicious vehicles are effectively off the network. The announcement can be performed by vehicle nodes or RSUs [Wan+20].

A method for detecting and preventing grayhole attacks, along with mathematical proofs and a SUMO simulation, based on RSU monitoring is presented in [Mal+23], which tries to find misbehaving vehicles that do not forward messages by looking at specific fields embedded in those messages.

In summary, Table 5.1 shows potential solutions to address the issues found in the ETSI ITS standards, including the solutions presented in Section 5.1, and the ones described in this section.

5.4 Conclusion

In this chapter, we analyzed the current landscape of security challenges in V2X communications, with respect to ETSI ITS standards, specifications and reports. We identified vulnerabilities that pose substantial risks to safety and user trust.

We presented several scenarios where these issues can be leveraged by malicious users in order to cause cyber-physical damages or chaotic situations that can compromise road safety. We highlight the coexistence of legacy vehicles and V2X-enabled vehicles during a transitional period, since this situation might cause some of the issues presented in this chapter and have severe consequences in case of attack.

In addition, we simulated some of the scenarios using SUMO and TraCI to quantify the impact. The simulation source code (SUMO configuration files and Python source code for TraCI) has been uploaded to GitHub [7Ro25].

While SUMO and TraCI were suitable for the current study, expanding the simulations with tools that natively support V2X communications and Cooperative Awareness Messages would enable broader coverage of the security and privacy issues identified in this work. Incorporating such features represents a promising direction for future research.

In general terms, with the way standards are currently specified, there is no possibility to prevent attacks. In some cases, the existing mechanisms result in an attack from the security perspective, DoS situation, or are difficult to implement in practice.

Looking ahead, several avenues for future research and development emerge as critical to enhance the robustness and reliability of V2X networks. With the rise of connected vehicles, shared vehicles and autonomous vehicles, an increase in research projects in this field is expected.

Table 5.1. Summary of solutions to the issues found.

Issue	Description	Solution
Replay attack	CAM and VAM have a 16-bit field `generationDeltaTime` that measures timestamp in milliseconds modulo 65536, so the same timestamp is valid after ~ 65 seconds.	Increase the bit-length of the field `generationDeltaTime` to prevent wrap-around after a small period of time.
Spoofing attack	Vehicles choose their own pseudonym, so an adversary can use other vehicles' pseudonyms.	Recent works explore cryptographic approaches [PSFK15] such as lightweight schemes [NHZG25], zero-knowledge proofs [Tao+24] and ring signatures [Mun+20] to balance authentication and privacy. However, these solutions face practical challenges, including limited scalability, vague benefits, and significant communication overhead.
Sybil attack	Vehicles choose their own pseudonym from a pool of owned pseudonyms, so adversaries can disguise as several entities with distinct pseudonyms.	Sybil attack detection techniques include radio frequency measurements [NSSP04; XYG06], RSU timestamps and trajectory analysis (such as RobSAD [CWHZ09]). Deterministic approaches have also been proposed, for example, embedding fields in pseudonym certificates to prevent overlaps [TB19].
Grayhole attack	Malicious vehicles can suppress DENM forwarding so that other vehicles are unaware of dangerous situations. The DCC mechanism can prevent honest vehicles from sending messages to the network.	Detection mechanisms use methods such as reputation, ACKs, or message analysis in vehicle nodes or RSUs [Wan+20]. Prevention involves announcing malicious IDs to isolate attackers [Mal+23].

Chapter 6

Conclusions, Contributions and Future Work

This chapter concludes the work presented. First, the conclusions and contributions are presented. Then, directions for future work are proposed. Finally, the original publications that resulted from this thesis and other works are listed.

6.1 Conclusions

In the comprehensive journey embodied in this doctoral thesis, we have analyzed several aspects related to automotive cybersecurity.

Chapter 1 provided background and motivation that make this research work valuable for the research community. After defining the scope and objectives of the thesis, we analyzed several vehicle subsystems, such as keyless entry systems, OBD-II dongles, and V2X communication standards.

Chapter 2 provided the foundational background and a comprehensive review of the state of the art that informed the rest of the thesis. This review contextualized the research by surveying existing work and established a common framework of terminology, standards and technical concepts. Finally, some research gaps were identified.

Chapter 3 presented a reverse-engineering methodology for analyzing RKE protocols, proving its effectiveness through the discovery of a vulnerability in the protocol implemented by a well-known car company that allowed to clone the key fob with a single signal capture. Using commercial hardware, open-source tools, and a black-box approach, the method enables quick auditing of RKE protocols without requiring costly specialized tools. Its validity was reinforced by results on another protocol (Volkswagen), which aligned with prior firmware-based studies.

Chapter 4 introduced a testing platform for OBD-II dongle cyberattacks, accompanied by the tool `pwnobd` [Nnu], and demonstrated how these resources improve researcher productivity for vulnerability analyses. The platform's usefulness was shown through a relevant case study (CVE-2016-2354 [CMU16]), evaluations of multiple devices within a basic test framework, and the public presentation of `pwnobd` at Black Hat Europe 2024 (Arsenal) [GG].

Chapter 5 examined security challenges in V2X communications within the framework of ETSI ITS standards, identifying critical vulnerabilities that threaten safety and user trust. Several attack scenarios were described and simulated with SUMO and TraCI to quantify their potential impact, emphasizing risks during the transitional period when legacy and V2X-enabled vehicles will coexist. A summary of potential mitigations was provided, though the analysis showed that current standards leave systems largely exposed, with some mechanisms themselves introducing DoS risks or being impractical to implement.

6.2 Contributions

The objectives of this doctoral thesis described in Section 1.2 have been successfully achieved:

1. Analyze existing cyberattacks and search for new attack vectors and vulnerabilities.

   This objective has been addressed during the state-of-the-art review, which is described in Chapter 2. We found new attack vectors that apply to RKE and PKES (Chapter 3) and OBD-II dongles that use Bluetooth to connect to the CAN bus (Chapter 4). This research work was published in [GLP25b].

2. Build a methodology for analyzing automotive subsystems from a black-box perspective.

   Chapter 3 and Chapter 4 tackle this objective with two different perspectives: RKE/PKES systems and OBD-II dongles. The described methodologies can be useful for future cybersecurity researchers that want to analyze these automotive subsystems or apply a similar procedure to any other subsystem.

3. Develop tools for auditing automotive subsystems.

   In Chapter 3, we found a vulnerability in one of the tested RKE models (Section 3.3.8) and developed a tool `rf_exploit` [7Ro24] to exploit it in order to clone the key fob. This research work was published in [GLP25a].

   Moreover, Chapter 4 describes a framework called `pwnobd` [Nnu] that enables cybersecurity researchers to interact with the CAN bus using OBD-II dongles connected with Bluetooth to an Android device. In fact, `pwnobd` was presented in Black Hat Europe 2024 (Arsenal) [GG], and the research work was published in [GGPL25].

4. Find security gaps in V2X communications protocols and standards for connected vehicles.

   Related to V2X communications, we analyzed ETSI ITS standards in Chapter 5, where we describe security issues and vulnerabilities found. In addition, we present several attack scenarios that can exploit these vulnerabilities to cause cyber-physical damages, and quantify their impact with simulations. This research work is currently under review in a journal [GYLP25].

In addition, the main objective of this thesis has also been addressed, since we have developed novel and flexible methodologies and tools to analyze the level of cybersecurity of nowadays and future vehicles and their related technologies. We have covered technologies that might be present in legacy vehicles (RKE protocols and OBD-II) and modern vehicles (PKES protocols and V2X communications). Thus, the implications of this research extend beyond academic boundaries, offering practical insights and tools for tackling urgent challenges related to automotive security and safety.

6.3 Future work

The automotive sector is in continuous growth and new technologies are being adopted in vehicular subsystems. As automotive manufacturers continue to integrate advanced technologies to improve safety, security, driving experience, and comfort, the complexity of automotive systems grows exponentially. These technologies introduce new attack vectors that cybercriminals can leverage. It is a fact that vehicles still lack cybersecurity best-practices. For instance, proprietary cryptographic protocols often exhibit weaknesses, infotainment systems are frequently exploited due to software vulnerabilities, and keyless entry systems remain susceptible to cyberattacks such as replay and relay attacks, making them easy targets for car thieves. In addition, the growing adoption of V2X communications further expands the attack surface. So, as new technologies are introduced, more cybersecurity analyses will be needed.

Regarding RKE/PKES systems, the proposed black-box methodology has certain limitations. For instance, it is difficult to distinguish between random bytes and encrypted bytes. Plus, since manufacturers use proprietary ciphers, it is almost impossible to guess the algorithm from this perspective. Some of the tested RKE models still need more reverse-engineering analysis to fully determine their frame structure. Working on reverse-engineering tools to analyze these protocols is a promising direction for future work. Another possible line of work is to develop RKE/PKES protocols that use standardized and known robust cryptographic algorithms, such as AES, and implement it on hardware devices. This will make RKE/PKES systems more secure and less vulnerable to the attacks presented in Chapter 2 and Chapter 3. In the wake of quantum computing, it will also be necessary to analyze if such protocols and algorithms are resistant to quantum attacks, especially considering the lifespan of vehicles.

In relation to OBD-II penetration testing with Bluetooth dongles and `pwnobd`, future work could focus on broadening the scope of tests to include a larger variety of devices, potentially leveraging established assessment frameworks such as IoTSF [IoT21] and BSAM [Tar24]. The platform `pwonbd` itself could be further enhanced, particularly by incorporating capabilities for Bluetooth connection interception and hijacking to simulate advanced attack scenarios. Additional software components and tighter integration between them could also be developed and evaluated to expand the platform's overall functionality. Finally, the proposed

architecture has the potential to be extended for the analysis of a wider range of devices across the broader IoT ecosystem.

With regards to V2X communication specification, in particular ETSI ITS standards, there are still security issues that need solution proposals that are really practical and implementable. Furthermore, although SUMO and TraCI proved adequate for the work presented in Chapter 5, using simulation tools that provide native support for V2X communications and Cooperative Awareness Messages would allow for more comprehensive assessment of the security issues identified. Integrating these capabilities represents a valuable avenue for future research.

In summary, this thesis has shown how several subsystems of a vehicle can be abused by cybercriminals to steal a car, steal personal belongings, obtain information, cause cyber-physical damages, or even control a car remotely. Thus, there is still work to be carried out to improve the security of these vehicular subsystems.

6.4 Original publications of the thesis

6.4.1 Articles published in peer-reviewed academic journals

- Roberto Gesteira-Miñarro, Gregorio López, and Rafael Palacios. "Revisiting Wireless Cyberattacks on Vehicles". *Sensors* 25.8 (Apr. 2025). ISSN: 1424-8220. DOI: 10.3390/s25082605.

  – JCR-JIF: 3,500 Q2 (2024) – SJR: 0,764 Q1 (2024).

- Roberto Gesteira-Miñarro, Gregorio López, and Rafael Palacios. "Clonable key fobs: Analyzing and breaking RKE protocols". *International Journal of Information Security* 24.3 (May 2025). ISSN: 1615-5262. DOI: 10.1007/s10207-025-01063-7.

  – JCR-JIF: 3,200 Q2 (2024) – SJR: 0,753 Q2 (2024).

- Roberto Gesteira-Miñarro, Ignacio Gutiérrez, Rafael Palacios, and Gregorio López. "pwnobd: Offensive Cybersecurity Toolkit for Vulnerability Analysis and Penetration Testing of OBD-II Devices". *IEEE Access* 13 (July 2025), pp. 126925–126934. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2025.3589867.

  – JCR-JIF: 3,600 Q2 (2024) – SJR: 0,849 Q1 (2024).

- [*Under review*] Roberto Gesteira-Miñarro, Takahito Yoshizawa, Gregorio López, and Rafael Palacios. "Highway to Hack – Security Gaps in ETSI ITS Standards". *Computer Standards & Interfaces* (2025). ISSN: 0920-5489.

  – JCR-JIF: 3.100 Q2 (2024) – SJR: 1.142 Q1 (2024).

6.4.2 Articles presented at academic conferences

- Roberto Gesteira-Miñarro. *Reverse-engineering radiofrequency protocols for Remote Keyless Entry systems*. Presented at conference: 19th Workshop on

Industrial Systems and Energy Technologies – JOSITE'2024. Instituto de Investigación Tecnológica. Universidad Pontificia Comillas, June 2024.

- Ignacio Gutiérrez, Gregorio López, Roberto Gesteira-Miñarro, and Rafael Palacios. "Plataforma de demostración para ataques extremo a extremo de dispositivos con interfaz OBD-II". *JNIC 2024: Actas de las IX Jornadas Nacionales de Investigación en Ciberseguridad*. May 2024, pp. 474–477. ISBN: 978-84-09-62140-8. URL: https://hdl.handle.net/11441/159179.

- Roberto Gesteira-Miñarro, Gregorio López, and Rafael Palacios. "Ingeniería inversa sobre protocolos de radiofrecuencia para sistemas Remote Keyless Entry". *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2023)*. June 2023, pp. 275–280. URL: http://hdl.handle.net/11093/4952.

6.4.3 Articles presented at international conferences

- Ignacio Gutiérrez and Roberto Gesteira-Miñarro. *pwnobd: Offensive cybersecurity toolkit for vulnerability analysis and penetration testing of OBD-II devices*. Black Hat Europe 2024 (Arsenal). URL: https://www.blackhat.com/eu-24/arsenal/schedule/index.html#pwnobd-offensive-cybersecurity-toolkit-for-vulnerability-analysis-and-penetration-testing-of-obd-ii-devices-42009.

6.5 Other publications during the thesis

6.5.1 Articles published in peer-reviewed academic journals

- Javier Jarauta Gastelu *et al.* "MitM Attack to Electric Vehicle AC Chargers". *IEEE Internet of Things Journal* (2025), pp. 1–11. ISSN: 2327-4662. DOI: 10.1109/JIOT.2025.3589219.

  – JCR-JIF: 8,900 Q1 (2024) – SJR: 2,483 Q1 (2024).

- [*Under review*] Clara Palacios-Castrillo *et al.* "Analysis of the Security and Privacy of Smart Personal Assistants with Real and Synthetic Voices". *Future Generation Computer Systems* (2025). ISSN: 1872-7115.

  – JCR-JIF: 6.100 Q1 (2024) – SJR: 1.551 Q1 (2024).

6.5.2 Articles presented at academic conferences

- Javier Jarauta *et al.* "Ataque MitM a puntos de recarga AC". *JNIC 2024: Actas de las IX Jornadas Nacionales de Investigación en Ciberseguridad*. May 2024, pp. 74–81. ISBN: 978-84-09-62140-8. URL: https://hdl.handle.net/11441/159179.

- Alejandro Manuel López Gómez *et al.* "Integración de un laboratorio de ciberseguridad OT en un laboratorio de automatización industrial". *JNIC 2024: Actas de las IX Jornadas Nacionales de Investigación en Ciberseguridad*. May 2024, pp. 134–140. ISBN: 978-84-09-62140-8. URL: https://hdl.handle.net/11441/159179.

- Clara Palacios-Castrillo, Rafael Palacios, Roberto Gesteira-Miñarro, and Gregorio López. "Análisis de seguridad y privacidad de asistentes personales con voces reales y voces sintéticas". *JNIC 2024: Actas de las IX Jornadas Nacionales de Investigación en Ciberseguridad*. May 2024, pp. 68–73. ISBN: 978-84-09-62140-8. URL: https://hdl.handle.net/11441/159179.

- José Antonio Font *et al.* "Threat models for vulnerability analysis of IoT devices for Manipulation of Demand attacks". *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2023)*. June 2023, pp. 573–580. URL: http://hdl.handle.net/11093/4952.

- Víctor García Fernández *et al.* "Dynamic risk assessment tool for IoT infrastructures for Smart Grids". *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2023)*. June 2023, pp. 363–366. URL: http://hdl.handle.net/11093/4952.

6.5.3 Articles presented at international conferences

- Javier Jarauta *et al. Ataque MitM a puntos de recarga AC*. RootedCON, Apr. 2024.

6.6 International research stay

To obtain the International Mention, the PhD candidate did a international research stay at KU Leuven (Belgium) for three months (17th February - 16th May 2025) in the Computer Science and Industrial Cryptography (COSIC) group, hosted by Prof. Bart Preneel supervised by Dr. Takahito Yoshizawa.

A relevant outcome of this international stay is a research paper in collaboration with Dr. Takahito Yoshizawaw:

- [*Under review*] Roberto Gesteira-Miñarro, Takahito Yoshizawa, Gregorio López, and Rafael Palacios. "Highway to Hack – Security Gaps in ETSI ITS Standards". *Computer Standards & Interfaces* (2025). ISSN: 0920-5489.
  - JCR-JIF: 3.100 Q2 (2024) – SJR: 1.142 Q1 (2024).

Bibliography

[5GA] 5GAA. *C-V2X explained*. 5GAA. URL: https://5gaa.org/c-v2x-explained.

[7Ro24] 7Rocky. ***rf_exploit***. 2024. URL: https://github.com/7Rocky/rf_exploit.

[7Ro25] 7Rocky. ***highway-to-hack***. 2025. URL: https://github.com/7Rocky/highway-to-hack.

[Abr24] Abraham, Ajin and MobSF collaborators. *Mobile-Security-Framework-MobSF*. Mar. 2024. URL: https://github.com/MobSF/Mobile-Security-Framework-MobSF.

[ACCM15] Oscar Alvear, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni. "Validation of a vehicle emulation platform supporting OBD-II communications". *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. Las Vegas, NV, USA: IEEE, Jan. 2015, pp. 880–885. ISBN: 978-1-4799-6390-4. DOI: 10.1109/CCNC.2015.7158092.

[AFRR23] Wasim A. Ali, Maria Pia Fanti, Michele Roccotelli, and Luigi Ranieri. "A Review of Digital Twin Technology for Electric and Autonomous Vehicles". *Applied Sciences* 13.1010 (Jan. 2023), p. 5871. ISSN: 2076-3417. DOI: 10.3390/app13105871.

[AK23] Shimaa A. Abdel Hakeem and Hyungwon Kim. "Authentication and encryption protocol with revocation and reputation management for enhancing 5G-V2X security". *Journal of King Saud University - Computer and Information Sciences* 35.7 (July 2023), p. 101638. ISSN: 1319-1578. DOI: 10.1016/j.jksuci.2023.101638.

[AM03] A.I. Alrabady and S.M. Mahmud. "Some attacks against vehicles' passive entry security systems and their solutions". *IEEE Transactions on Vehicular Technology* 52.2 (Mar. 2003), pp. 431–439. ISSN: 0018-9545. DOI: 10.1109/TVT.2003.808759.

[ASJ19] Aljawharah Alnasser, Hongjian Sun, and Jing Jiang. "Cyber Security Challenges and Solutions for V2X Communications: A Survey". *Computer Networks* 151 (Mar. 2019), pp. 1–48. DOI: 10.1016/j.comnet.2018.12.018.

[ASR] ASRG. *Enabling positive Security solutions for the Automotive Market*. ASRG. URL: https://asrg.io.

[atl] atlas0fd00m. ***rfcat***. URL: https://github.com/atlas0fd00m/rfcat.

[Bis] Bishop Fox and Sliver contributors. ***sliver***. URL: https://github.com/BishopFox/sliver.

[BKS24] Alexandra Boldyreva, Virendra Kumar, and Jiahao Sun. "Provable Security Analysis of Butterfly Key Mechanism Protocol in IEEE 1609.2.1 Standard". *Proceedings on Privacy Enhancing Technologies* 2024 (Oct. 2024), pp. 565–582. DOI: 10.56553/popets-2024-0130.

[BL16] Stefan van de Beek and Frank Leferink. "Vulnerability of Remote Keyless-Entry Systems Against Pulsed Electromagnetic Interference and Possible Improvements". *IEEE Transactions on Electromagnetic Compatibility* 58.4 (Aug. 2016), pp. 1259–1265. ISSN: 1558-187X. DOI: 10.1109/TEMC.2016.2570303.

[Blu18] BlueDriverSupport. *Reply to "Friendly PSA: BlueDriver OBD2 Scanner Company (Lemur Vehicle Monitors) Seemingly Pulled from Market"*. Nov. 2018. URL: www.reddit.com/r/MechanicAdvice/comments/9t6e7m/friendly_psa_bluedriver_obd2_scanner_company/e8ujm1h/.

[BMR21] Ghanishtha Bhatti, Harshit Mohan, and R. Raja Singh. "Towards the future of smart electric vehicles: Digital twin technology". *Renewable and Sustainable Energy Reviews* 141 (May 2021), p. 110801. ISSN: 1364-0321. DOI: 10.1016/j.rser.2021.110801.

[Bog07] Andrey Bogdanov. "Cryptanalysis of the KeeLoq block cipher". *IACR Cryptol. ePrint Arch.* 2007 (Jan. 2007), p. 55. URL: https://eprint.iacr.org/2007/055.pdf.

[Bre+18] Benedikt Brecht *et al.* "A Security Credential Management System for V2X Communications". *IEEE Transactions on Intelligent Transportation Systems* 19.12 (2018), pp. 3850–3871. DOI: 10.1109/TITS.2018.2797529.

[BRLK17] Ryad Benadjila, Mathieu Renard, José Lopes-Esteves, and Chaouki Kasmi. "One Car, Two Frames: Attacks on Hitag-2 Remote Keyless Entry Systems Revisited". *Proceedings of the 11th USENIX Conference on Offensive Technologies*. WOOT'17. Vancouver, BC, Canada: USENIX Association, 2017. URL: https://www.usenix.org/system/files/conference/woot17/woot17-paper-benadjila.pdf.

[BSJ18] Mehmet Bozdal, Mohammad Samie, and Ian Jennions. "A Survey on CAN Bus Protocol: Attacks, Challenges, and Potential Solutions". *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. Southend, United Kingdom: IEEE, Aug. 2018, pp. 201–205. ISBN: 978-1-5386-4904-6. DOI: 10.1109/iCCECOME.2018.8658720.

[Che+11] Stephen Checkoway *et al.* "Comprehensive Experimental Analyses of Automotive Attack Surfaces". *20th USENIX Security Symposium (USENIX Security 11)*. San Francisco, CA, USA: USENIX Association, Aug. 2011, p. 6. DOI: 10.5555/2028067.2028073.

[Che+18] Jiongyi Chen *et al.* "IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing". *Proceedings 2018 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2018. ISBN: 978-1-891562-49-5. DOI: 10.14722/ndss.2018.23159.

[Cho+20] Abdullahi Chowdhury *et al.* "Attacks on Self-Driving Cars and Their Countermeasures: A Survey". *IEEE Access* 8 (2020), pp. 207308–207342. DOI: 10.1109/ACCESS.2020.3037705.

[CMU16] CMU. *VU#615456 - Lemur Vehicle Monitors BlueDriver LSB2 does not authenticate users for Bluetooth access*. Carnagie Mellon University, Apr. 2016. URL: https://www.kb.cert.org/vuls/id/615456.

[Coo24] Gregory Cook. *CRC RevEng: arbitrary-precision CRC calculator and algorithm finder*. 2024. URL: https://reveng.sourceforge.io.

[Csi+22] Levente Csikor *et al.* "RollBack: A New Time-Agnostic Replay Attack Against the Automotive Remote Keyless Entry Systems". *ACM Transactions on Cyber-Physical Systems* (2022). DOI: 10.48550/arXiv.2210.11923.

[CWHZ09] Chen Chen, Xin Wang, Weili Han, and Binyu Zang. "A Robust Detection of the Sybil Attack in Urban VANETs". *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*. 2009, pp. 270–276. DOI: 10.1109/ICDCSW.2009.48.

[Dav23] Nahla Davies. *The top 8 Cybersecurity threats facing the automotive industry heading into 2023*. LevelBlue, Feb. 2023. URL: https://cybersecurity.att.com/blogs/security-essentials/the-top-8-cybersecurity-threats-facing-the-automotive-industry-heading-into-2023.

[DERL22] Ahmed Didouh, Yassin El Hillali, Atika Rivenq, and Houda Labiod. "Novel Centralized Pseudonym Changing Scheme for Location Privacy in V2X Communication". *Energies* 15.3 (2022), pp. 1–18. ISSN: 1996-1073. DOI: 10.3390/en15030692.

[Dro15] Alex Drozhzhin. *Black Hat USA 2015: The full story of how that Jeep was hacked*. Aug. 2015. URL: https://www.kaspersky.com/blog/blackhat-jeep-cherokee-hack-explained/9493/.

[DS24] Jeremy Daily and Martin (Trae) Span. "A Model for Cybersecurity Education through Challenge Events". *INCOSE International Symposium* 34.1 (2024), pp. 944–957. ISSN: 2334-5837. DOI: 10.1002/iis2.13187.

[Eis+08a] Thomas Eisenbarth *et al.* "On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme". *Advances in Cryptology – CRYPTO 2008*. Berlin, Heidelberg: Springer Berlin Heidelberg, Aug. 2008, pp. 203–220. ISBN: 978-3-540-85173-8. DOI: 10.1007/978-3-540-85174-5_12.

[Eis+08b] Thomas Eisenbarth *et al.* "Physical Cryptanalysis of KeeLoq Code Hopping Applications". *IACR Cryptology ePrint Archive* 2008 (Jan. 2008). URL: https://eprint.iacr.org/2008/058.pdf.

[El-+20] Zeinab El-Rewini *et al.* "Cybersecurity challenges in vehicular communications". *Vehicular Communications* 23 (2020), p. 100214. ISSN: 2214-2096. DOI: 10.1016/j.vehcom.2019.100214.

[ELM10] ELM Electronics. *ELM327 - OBD to RS232 Interpreter*. 2010. URL: https://www.elmelectronics.com/DSheets/ELM327DSH.pdf.

[ELM18] ELM Electronics. *AN07 - Sending Arbitrary CAN Messages*. Version rev A. 2018. URL: https://www.elmelectronics.com/AppNotes/AppNote07.pdf.

[End10] Robert R. Enderlein. *KeeLoq*. Tech. rep. École Polytechnique Fédérale de Lausanne, Jan. 2010. URL: http://www.e7n.ch/data/e10.pdf.

[Era24] Danilo Erazo. *Breaking Learning Codes*. DEF CON 32, Aug. 2024. URL: https://media.defcon.org/DEF%20CON%2032/DEF%20CON%2032%20villages/DEF%20CON%2032%20-%20Car%20Hacking%20Village%20-%20Danilo%20Erazo%20-%20How%20I%20discovered%20and%20hacked%20Learning%20Codes%20of%20the%20key%20job%20of%20a%20car%20assembled%20in%20my%20country.pdf.

[ETS10] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Communications Architecture*. Tech. rep. ETSI EN 302 665 V1.1.1. Sept. 2010.

[ETS18a] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part*. Tech. rep. ETSI TS 102 687 V1.2.1. Apr. 2018.

[ETS18b] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Security; Pre-standardization study on pseudonym change management*. Tech. rep. ETSI TR 103 415 V1.1.1. Apr. 2018.

[ETS18c] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary*. Tech. rep. ETSI TS 102 894-2 V1.3.1. Aug. 2018.

[ETS19a] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*. Tech. rep. ETSI EN 302 637-3 V1.3.1. Apr. 2019.

[ETS19b] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*. Tech. rep. ETSI EN 302 637-2 V1.4.1. Apr. 2019.

[ETS20] European Telecommunications Standards Institute (ETSI). *Intelligent Transport System (ITS); Vulnerable Road Users (VRU) awareness; Part 2: Functional Architecture and Requirements definition; Release 2*. Tech. rep. ETSI TS 103 300-2 V2.1.1. May 2020.

[ETS21] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Security; Security header and certificate formats; Release 2*. Tech. rep. ETSI TS 103 097 V2.1.1. Oct. 2021.

[ETS22] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Security; Trust and Privacy Management; Release 2*. Tech. rep. ETSI TS 102 941 V2.2.1. Nov. 2022.

[ETS23] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Vulnerable Road Users (VRU) awareness; Part 3: Specification of VRU awareness basic service; Release 2*. Tech. rep. ETSI TS 103 300-3 V2.2.1. Feb. 2023.

[ETS24] European Telecommunications Standards Institute (ETSI). *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Release 2*. Tech. rep. ETSI TR 102 638 V2.1.1. Apr. 2024.

[Fan+16] Qing Fan *et al.* "VANET Routing Replay Attack Detection Research Based on SVM". *MATEC Web of Conferences* 63 (Jan. 2016), p. 05020. DOI: 10.1051/matecconf/20166305020.

[FDC11] Aurélien Francillon, Boris Danev, and Srdjan Capkun. "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars". *IACR Cryptol. ePrint Arch.* (2011). DOI: 10.3929/ETHZ-A-006708714.

[Fen+21] Xiaotao Feng *et al.* "Snipuzz: Black-box Fuzzing of IoT Firmware via Message Snippet Inference". *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. CCS '21. ACM, Nov. 2021. DOI: 10.1145/3460120.3484543.

[FGSS18] Renato Ferreira, João Gaspar, Nuno Souto, and Pedro Sebastião. "Effective GPS Jamming Techniques for UAVs Using Low-Cost SDR Platforms". *2018 Global Wireless Summit (GWS)*. 2018, pp. 27–32. DOI: 10.1109/GWS.2018.8686672.

[Fon+23] José Antonio Font *et al.* "Threat models for vulnerability analysis of IoT devices for Manipulation of Demand attacks". *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2023)*. June 2023, pp. 573–580. URL: http://hdl.handle.net/11093/4952.

[GA22] Ahmed Ghanem and Riham AlTawy. "Garage Door Openers: A Rolling Code Protocol Case Study". *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*. Aug. 2022, pp. 1–6. DOI: 10.1109/PST55820.2022.9851991.

[Gad24a] Great Scott Gadgets. *HackRF One*. Great Scott Gadgets, 2024. URL: https://greatscottgadgets.com/hackrf/one/.

[Gad24b] Great Scott Gadgets. *YARD Stick One*. Great Scott Gadgets, 2024. URL: https://greatscottgadgets.com/yardstickone/.

[Gar+23] Víctor García Fernández *et al.* "Dynamic risk assessment tool for IoT infrastructures for Smart Grids". *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2023)*. June 2023, pp. 363–366. URL: http://hdl.handle.net/11093/4952.

[Gas+25] Javier Jarauta Gastelu *et al.* "MitM Attack to Electric Vehicle AC Chargers". *IEEE Internet of Things Journal* (2025), pp. 1–11. ISSN: 2327-4662. DOI: 10.1109/JIOT.2025.3589219.

[Ges24] Roberto Gesteira-Miñarro. *Reverse-engineering radiofrequency protocols for Remote Keyless Entry systems*. Presented at conference: 19th Workshop on Industrial Systems and Energy Technologies – JOSITE'2024. Instituto de Investigación Tecnológica. Universidad Pontificia Comillas, June 2024.

[GG] Ignacio Gutiérrez and Roberto Gesteira-Miñarro. *pwnobd: Offensive cybersecurity toolkit for vulnerability analysis and penetration testing of OBD-II devices*. Black Hat Europe 2024 (Arsenal). URL: https://www.blackhat.com/eu-24/arsenal/schedule/index.html#pwnobd-offensive-cybersecurity-toolkit-for-vulnerability-analysis-and-penetration-testing-of-obd-ii-devices-42009.

[GGPL25] Roberto Gesteira-Miñarro, Ignacio Gutiérrez, Rafael Palacios, and Gregorio López. "pwnobd: Offensive Cybersecurity Toolkit for Vulnerability Analysis and Penetration Testing of OBD-II Devices". *IEEE Access* 13 (July 2025), pp. 126925–126934. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2025.3589867.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. 28th printing. Addison-Wesley professional computing series. Reading, MA: Addison-Wesley, 1995. ISBN: 978-0-201-63361-0. URL: https://core.ac.uk/download/pdf/25185422.pdf.

[GHM23] Mansi Girdhar, Junho Hong, and John Moore. “Cybersecurity of Autonomous Vehicles: A Systematic Literature Review of Adversarial Attacks and Defense Models”. *IEEE Open Journal of Vehicular Technology* 4 (2023), pp. 417–437. DOI: 10.1109/OJVT.2023.3265363.

[GLGP24] Ignacio Gutiérrez, Gregorio López, Roberto Gesteira-Miñarro, and Rafael Palacios. “Plataforma de demostración para ataques extremo a extremo de dispositivos con interfaz OBD-II”. *JNIC 2024: Actas de las IX Jornadas Nacionales de Investigación en Ciberseguridad*. May 2024, pp. 474–477. ISBN: 978-84-09-62140-8. URL: https://hdl.handle.net/11441/159179.

[GLP23] Roberto Gesteira-Miñarro, Gregorio López, and Rafael Palacios. “Ingeniería inversa sobre protocolos de radiofrecuencia para sistemas Remote Keyless Entry”. *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2023)*. June 2023, pp. 275–280. URL: http://hdl.handle.net/11093/4952.

[GLP25a] Roberto Gesteira-Miñarro, Gregorio López, and Rafael Palacios. “Clonable key fobs: Analyzing and breaking RKE protocols”. *International Journal of Information Security* 24.3 (May 2025). ISSN: 1615-5262. DOI: 10.1007/s10207-025-01063-7.

[GLP25b] Roberto Gesteira-Miñarro, Gregorio López, and Rafael Palacios. “Revisiting Wireless Cyberattacks on Vehicles”. *Sensors* 25.8 (Apr. 2025). ISSN: 1424-8220. DOI: 10.3390/s25082605.

[GME17] Tobias Glocker, Timo Mantere, and Mohammed Elmusrati. “A protocol for a secure remote keyless entry system applicable in vehicles using symmetric-key cryptography”. *2017 8th International Conference on Information and Communication Systems (ICICS)*. Apr. 2017, pp. 310–315. DOI: 10.1109/IACS.2017.7921990.

[GOKP16] Flavio D. Garcia, David Oswald, Timo Kasper, and Pierre Pavlidès. “Lock It and Still Lose It - on the (in)Security of Automotive Remote Keyless Entry Systems”. *Proceedings of the 25th USENIX Conference on Security Symposium*. SEC’16. Austin, TX, USA: USENIX Association, 2016, pp. 929–944. ISBN: 978-1-931971-324. URL: https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_garcia.pdf.

[Gre+20] Kyle Greene *et al.* “Timestamp-based Defense Mechanism Against Replay Attack in Remote Keyless Entry Systems”. *2020 IEEE International Conference on Consumer Electronics (ICCE)*. Jan. 2020, pp. 1–4. DOI: 10.1109/ICCE46568.2020.9043039.

[Gre15] Andy Greenberg. *Hackers Remotely Kill a Jeep on the Highway—With Me in It*. July 2015. URL: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[GYLP25] Roberto Gesteira-Miñarro, Takahito Yoshizawa, Gregorio López, and Rafael Palacios. "Highway to Hack – Security Gaps in ETSI ITS Standards". *Computer Standards & Interfaces* (2025). ISSN: 0920-5489.

[Ham+17] Badis Hammi *et al.* "Using butterfly keys: A performance study of pseudonym certificates requests in C-ITS". *2017 1st Cyber Security in Networking Conference (CSNet)*. 2017, pp. 1–6. DOI: 10.1109/CSNET.2017.8242002.

[Ham+22] Badis Hammi *et al.* "Is it Really Easy to Detect Sybil Attacks in C-ITS Environments: A Position Paper". *IEEE Transactions on Intelligent Transportation Systems* 23.10 (2022), pp. 18273–18287. DOI: 10.1109/TITS.2022.3165513.

[HCA+11] Eric M Hutchins, Michael J Cloppert, Rohan M Amin, *et al.* "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains". *Leading Issues in Information Warfare & Security Research* 1.1 (2011), p. 80.

[He+23] Chao He *et al.* "Security and Privacy in Vehicular Digital Twin Networks: Challenges and Solutions". *IEEE Wireless Communications* 30.4 (Aug. 2023), pp. 154–160. ISSN: 1558-0687. DOI: 10.1109/MWC.002.2200015.

[HGO18] Christopher Hicks, Flavio D. Garcia, and David Oswald. "Dismantling the AUT64 Automotive Cipher". *IACR Transactions on Cryptographic Hardware and Embedded Systems* (May 2018), pp. 46–69. ISSN: 2569-2925. DOI: 10.13154/tches.v2018.i2.46-69.

[HNBS25] Quanrui Huo, Yuqiao Ning, Chenya Bian, and Dongqing Sun. "Research on Anti-Replay Attack Mechanism of Intelligent Connected Vehicles Based on Hashing Chain and V2X Communication". *The International Conference Optoelectronic Information and Optical Engineering (OIOE2024)*. Ed. by Yang Yue and Lu Leng. Vol. 13513. International Society for Optics and Photonics. SPIE, 2025, 135133H. DOI: 10.1117/12.3054402.

[Hon16] Marshall Honorof. *This Bluetooth Car Dongle Might Just Kill You.* Apr. 2016. URL: https://www.tomsguide.com/us/lemur-bluedriver-security-flaw,news-22521.html.

[HS25] Yuki Higashida and Kenya Sato. "Evaluation of a Pseudonym Change Scheme using LSH for Location Privacy in V2X Communication". *IEICE Communications Express* 14.3 (2025), pp. 115–118. DOI: 10.23919/comex.2024XBL0184.

[Hu+22] Zhongxu Hu *et al.* "Review and Perspectives on Driver Digital Twin and Its Enabling Technologies for Intelligent Vehicles". *IEEE Transactions on Intelligent Vehicles* 7.3 (Sept. 2022), pp. 417–440. ISSN: 2379-8904. DOI: 10.1109/TIV.2022.3195635.

[HZB18] Tianxiang Huang, Jianying Zhou, and Andrei Bytes. "ATG: An Attack Traffic Generation Tool for Security Testing of In-vehicle CAN Bus". *Proceedings of the 13th International Conference on Availability, Reliability and Security*. Hamburg Germany: ACM, Aug. 2018, pp. 1–6. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3230843.

[IHOD19] Omar Adel Ibrahim, Ahmed Mohamed Hussain, Gabriele Oligeri, and Roberto Di Pietro. "Key is in the Air: Hacking Remote Keyless Entry Systems". *Security and Safety Interplay of Intelligent Software Systems*. Ed. by Brahim Hamid *et al.* Cham: Springer International Publishing, 2019, pp. 125–132. ISBN: 978-3-030-16874-2. DOI: 10.1007/978-3-030-16874-2_9.

[Imm12] Vincent Immler. "Breaking Hitag 2 Revisited". *Proceedings of the Second International Conference on Security, Privacy, and Applied Cryptography Engineering*. SPACE'12. Chennai, India: Springer-Verlag, 2012, pp. 126–143. ISBN: 978-3-642344-152. DOI: 10.1007/978-3-642-34416-9_9.

[Ind+08] Sebastiaan Indesteege *et al.* "A Practical Attack on KeeLoq". *Advances in Cryptology – EUROCRYPT 2008*. Ed. by Nigel Smart. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–18. ISBN: 978-3-540-78967-3. DOI: 10.1007/978-3-540-78967-3_1.

[Int22] SAE International. *V2X Communications Message Set Dictionary*. Tech. rep. J2735. 2022.

[Int24] International Organization for Standardization. *Road vehicles – Diagnostic communication over Controller Area Network (DoCAN)*. Tech. rep. ISO 15765-2:2024. 2024. URL: https://www.iso.org/standard/84211.html.

[Int94] International Organization for Standardization. *Road vehicles – Diagnostic systems*. Tech. rep. ISO 9141-2:1994. 1994. URL: https://www.iso.org/standard/16738.html.

[IoT21] IoTSF. *IoT Security Assurance Framework Release 3.0*. Nov. 2021. URL: https://iotsecurityfoundation.org/wp-content/uploads/2021/11/IoTSF-IoT-Security-Assurance-Framework-Release-3.0-Nov-2021-1.pdf.

[Jar+24a] Javier Jarauta *et al.* "Ataque MitM a puntos de recarga AC". *JNIC 2024: Actas de las IX Jornadas Nacionales de Investigación en Ciberseguridad*. May 2024, pp. 74–81. ISBN: 978-84-09-62140-8. URL: https://hdl.handle.net/11441/159179.

[Jar+24b] Javier Jarauta *et al. Ataque MitM a puntos de recarga AC*. RootedCON, Apr. 2024.

[JCL20] Kyungho Joo, Wonsuk Choi, and Dong Hoon Lee. “Hold the Door! Fingerprinting Your Car Key to Prevent Keyless Entry Car Theft”. *Proceedings 2020 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2020. ISBN: 978-1-891562-61-7. DOI: 10.14722/ndss.2020.23107.

[Kam15] Samy Kamkar. *Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars*. DEF CON 23, Aug. 2015. URL: https://defcon.org/html/defcon-23/dc-23-speakers.html#Kamkar.

[Kan+21a] Hyunjae Kang *et al.* “Car Hacking and Defense Competition on In-Vehicle Network”. *Proceedings Third International Workshop on Automotive and Autonomous Vehicle Security*. AutoSec 2021. Internet Society, 2021. DOI: 10.14722/autosec.2021.23035.

[Kan+21b] Hyunjae Kang *et al. Car Hacking: Attack & Defense Challenge 2020 Dataset*. 2021. DOI: 10.21227/qvr7-n418.

[Ken24] Sami Alaoui Kendil. *Reverse engineering a car key fob signal (Part 1)*. Mar. 2024. URL: https://0x44.cc/radio/2024/03/13/reversing-a-car-key-fob-signal.html.

[KKMP09] Markus Kasper, Timo Kasper, Amir Moradi, and Christof Paar. “Breaking KeeLoq in a Flash: On Extracting Keys at Lightning Speed”. *Progress in Cryptology – AFRICACRYPT 2009*. Ed. by Bart Preneel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 403–420. ISBN: 978-3-642-02384-2. DOI: 10.1007/978-3-642-02384-2_25.

[Kos+10] Karl Koscher *et al.* “Experimental Security Analysis of a Modern Automobile”. *2010 IEEE Symposium on Security and Privacy*. Oakland, CA, USA: IEEE, 2010, pp. 447–462. ISBN: 978-1-4244-6894-2. DOI: 10.1109/SP.2010.34.

[KR24] Md Rafiul Kabir and Sandip Ray. “ViSE: Digital Twin Exploration for Automotive Functional Safety and Cybersecurity”. *Journal of Hardware and Systems Security* (May 2024). ISSN: 2509-3428, 2509-3436. DOI: 10.1007/s41635-024-00150-w.

[Lad21] Kartheek Lade. *CANalyse*. June 2021. URL: https://github.com/KartheekLade/CANalyse.

[LH23] Aaron Luo and Spencer Hsieh. *Remotely Hacking a car through an OBD-II Bluetooth Dongle*. Automotive Security Research Group, July 2023. URL: https://www.youtube.com/watch?v=f19_BNgVrWQ.

[Li+23] Tianyi Li *et al.* “Exploring Energy Impacts of Cyberattacks on Adaptive Cruise Control Vehicles”. *2023 IEEE Intelligent Vehicles Symposium (IV)*. 2023, pp. 1–6. DOI: 10.1109/IV55152.2023.10186730.

[Lia] Lianghung Co. *OBD-II ECU Simulator CAN BUS*. URL: https://vehelec.com/products/obd-ii-ecu-simulator-iso15765-iso9141-2-iso14230-can-bus.

[LM15] Lan Lin and James A. Misener. “Message Sets for Vehicular Communications”. *Vehicular ad hoc Networks* (Jan. 2015), pp. 123–163. DOI: 10.1007/978-3-319-15497-8_5.

[LN24] Govindarajan Lakshmikanthan and Sreejith Sreekandan Nair. “Mitigating Replay Attacks in Autonomous Vehicles”. *International Research Journal of Engineering and Technology* 11 (May 2024), pp. 1–7.

[Lon+23] Brigitte Lonc *et al.* “Feasibility and Benchmarking of Post-Quantum Cryptography in the Cooperative ITS Ecosystem”. *2023 IEEE Vehicular Networking Conference (VNC)*. Apr. 2023, pp. 215–222. DOI: 10.1109/VNC57357.2023.10136335.

[Lop+18] Pablo Alvarez Lopez *et al.* “Microscopic Traffic Simulation using SUMO”. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2575–2582. DOI: 10.1109/ITSC.2018.8569938.

[Lóp+24] Alejandro Manuel López Gómez *et al.* “Integración de un laboratorio de ciberseguridad OT en un laboratorio de automatización industrial”. *JNIC 2024: Actas de las IX Jornadas Nacionales de Investigación en Ciberseguridad*. May 2024, pp. 134–140. ISBN: 978-84-09-62140-8. URL: https://hdl.handle.net/11441/159179.

[LSWS25] Tianyi Li, Mingfeng Shang, Shian Wang, and Raphael Stern. “Detecting Subtle Cyberattacks on Adaptive Cruise Control Vehicles: A Machine Learning Approach”. *IEEE Open Journal of Intelligent Transportation Systems* 6 (2025), pp. 11–23. DOI: 10.1109/OJITS.2024.3522969.

[Mal+23] Abdul Malik *et al.* “An Efficient Approach for the Detection and Prevention of Gray-Hole Attacks in VANETs”. *IEEE Access* 11 (2023), pp. 46691–46706. DOI: 10.1109/ACCESS.2023.3274650.

[mie] miek. *inspectrum*. URL: https://github.com/miek/inspectrum.

[MK09] Amir Moradi and Timo Kasper. “A new remote keyless entry system resistant to power analysis attacks”. *2009 7th International Conference on Information, Communications and Signal Processing (ICICS)*. Dec. 2009, pp. 1–6. DOI: 10.1109/ICICS.2009.5397727.

[Mun+20] Pravin Mundhe *et al.* “Ring Signature-Based Conditional Privacy-Preserving Authentication in VANETs”. *Wirel. Pers. Commun.* 114.1 (Sept. 2020), pp. 853–881. ISSN: 0929-6212. DOI: 10.1007/s11277-020-07396-x.

[MV15] Charlie Miller and Chris Valasek. *Remote Exploitation of an Unaltered Passenger Vehicle*. Tech. rep. IOActive, 2015. URL: https://www.ioactive.com/wp-content/uploads/pdfs/IOActive_Remote_Car_Hacking.pdf.

[NHZG25] Sujash Naskar, Gerhard Hancke, Tingting Zhang, and Mikael Gidlund. "Pseudo-Random Identification and Efficient Privacy-Preserving V2X Communication for IoV Networks". *IEEE Access* 13 (2025), pp. 1147–1163. DOI: 10.1109/ACCESS.2024.3523358.

[NIS22] NIST. *Lightweight Cryptography*. NIST, Jan. 2022. URL: https://www.nist.gov/programs-projects/lightweight-cryptography.

[Nnu] Nnubes256. *pwnobd*. URL: https://github.com/Nnubes256/pwnobd.

[NSSP04] J. Newsome, E. Shi, D. Song, and A. Perrig. "The Sybil attack in sensor networks: analysis & defenses". *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004.* 2004, pp. 259–268. DOI: 10.1145/984622.984660.

[Oss] Michael Ossmann. *Rapid Radio Reversing*. Black Hat Asia 2016. URL: https://www.blackhat.com/docs/asia-16/materials/asia-16-Ossmann-Rapid-Radio-Reversing-wp.pdf.

[OZ924] Alexandru Csete OZ9AEC. *Gqrx SDR - Open source software defined radio by Alexandru Csete OZ9AEC*. Gqrx SDR, 2024. URL: https://gqrx.dk/.

[Pal+25] Clara Palacios-Castrillo *et al.* "Analysis of the Security and Privacy of Smart Personal Assistants with Real and Synthetic Voices". *Future Generation Computer Systems* (2025). ISSN: 1872-7115.

[Pal23] Zac Palmer. *Thieves are now stealing cars via a headlight 'CAN injection'*. Apr. 2023. URL: https://www.autoblog.com/carbuying/vehicle-headlight-can-bus-injection-theft-method-update.

[PAS21] Simon Grønfeldt Philipsen, Birger Andersen, and Bhupjit Singh. "Threats and Attacks to Modern Vehicles". *2021 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*. Nov. 2021, pp. 22–27. DOI: 10.1109/IoTaIS53735.2021.9628576.

[PAZ13] Soyoung Park, Baber Aslam, and Cliff Zou. "Defense against Sybil attack in the initial deployment stage of vehicular ad hoc network based on roadside unit support". *Security and Communication Networks* 6 (Apr. 2013). DOI: 10.1002/sec.679.

[PM23] Ramiro Pareja Veredas and Yashin Mehaboobe. "Attacking Vehicle Fleet Management Systems". Presented at conference: Hardwear.io NL 2023. 2023. URL: https://www.youtube.com/watch?v=FycVqUvicJw.

[PN18] Johannes Pohl and Andreas Noack. "Universal Radio Hacker: A Suite for Analyzing and Attacking Stateful Wireless Protocols". *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. Baltimore, MD: USENIX Association, 2018. URL: https://www.usenix.org/conference/woot18/presentation/pohl.

[PN24] Alexey Zakharov Pavel Zhovner and Ruslan Nadyrshin. *Our Response to the Canadian Government*. Flipper, Mar. 2024. URL: https://blog.flipper.net/response-to-canadian-government/.

[PPGL24] Clara Palacios-Castrillo, Rafael Palacios, Roberto Gesteira-Miñarro, and Gregorio López. "Análisis de seguridad y privacidad de asistentes personales con voces reales y voces sintéticas". *JNIC 2024: Actas de las IX Jornadas Nacionales de Investigación en Ciberseguridad*. May 2024, pp. 68–73. ISBN: 978-84-09-62140-8. URL: https://hdl.handle.net/11441/159179.

[Pro] Proxmark. *Proxmark 3 RDV4*. URL: https://proxmark.com/proxmark-3-hardware/proxmark-3-rdv4.

[PS15] Jonathan Petit and Steven E. Shladover. "Potential Cyberattacks on Automated Vehicles". *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2015), pp. 546–556. DOI: 10.1109/TITS.2014.2342271.

[PS22] Rohini Poolat Parameswarath and Biplab Sikdar. "An Authentication Mechanism for Remote Keyless Entry Systems in Cars to Prevent Replay and RollJam Attacks". *2022 IEEE Intelligent Vehicles Symposium (IV)*. June 2022, pp. 1725–1730. DOI: 10.1109/IV51971.2022.9827256.

[PSFK15] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. "Pseudonym Schemes in Vehicular Networks: A Survey". *IEEE Communications Surveys & Tutorials* 17 (Mar. 2015), pp. 228–255. DOI: 10.1109/COMST.2014.2345420.

[PX21] Minh Pham and Kaiqi Xiong. "A survey on security attacks and defense techniques for connected and autonomous vehicles". *Computers & Security* 109 (Oct. 2021), p. 102269. ISSN: 0167-4048. DOI: 10.1016/j.cose.2021.102269.

[Rad24] GNU Radio. *About GNU Radio*. GNU Radio, 2024. URL: https://www.gnuradio.org/about/.

[Rap24a] Rapid7. *Hardware Bridge Session Connector*. 2024. URL: https://www.rapid7.com/db/modules/auxiliary/client/hwbridge/connect/.

[Rap24b] Rapid7, Inc. and Metasploit Framework contributors. *Metasploit Framework*. June 2024. URL: https://github.com/rapid7/metasploit-framework.

[RG20] Andreea-Ina Radu and Flavio D. Garcia. "Grey-box Analysis and Fuzzing of Automotive Electronic Components via Control-Flow Graph Extraction". *Proceedings of the 4th ACM Computer Science in Cars Symposium*. CSCS '20. Feldkirchen, Germany: Association for Computing Machinery, 2020. ISBN: 978-1-450376-211. DOI: 10.1145/3385958.3430480.

[Rob16] Paul Roberts. *CERT: Aftermarket Add-On Opens Cars To Life Threatening Hacks*. Apr. 2016. URL: https://securityledger.com/2016/04/cert-warns-on-hacking-risk-to-bluedriver-plug-in/.

[Rou+10] Ishtiaq Rouf *et al.* "Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study". *Proceedings of the 19th USENIX Conference on Security*. USENIX Security'10. Washington, DC: USENIX Association, 2010, p. 21. URL: https://www.usenix.org/legacy/events/sec10/tech/full_papers/Rouf.pdf.

[SHKL22] Rajkumar Singh Rathore, Chaminda Hewage, Omprakash Kaiwartya, and Jaime Lloret. "In-Vehicle Communication Cyber Security: Challenges and Solutions". *Sensors* (Sept. 2022). DOI: 10.3390/s22176679.

[Smi17] Craig Smith. "Latest Metasploit Hardware Bridge Techniques". Presented at conference: Hardwear.io 2017. 2017. URL: https://www.youtube.com/watch?v=eqIIi7AZfOI.

[SMP16] Iraklis Symeonidis, Mustafa A. Mustafa, and Bart Preneel. "Keyless car sharing system: A security and privacy analysis". *2016 IEEE International Smart Cities Conference (ISC2)*. Sept. 2016, pp. 1–7. DOI: 10.1109/ISC2.2016.7580758.

[SNMB10] Nabil Seddigh, Biswajit Nandy, Rupinder Makkar, and Jean-Francois Beaumont. "Security Advances and Challenges in 4G Wireless Networks". *2010 Eighth International Conference on Privacy, Security and Trust*. Aug. 2010, pp. 62–71. DOI: 10.1109/PST.2010.5593244.

[SYZ22] Xiaoqiang Sun, F. Richard Yu, and Peng Zhang. "A Survey on Cyber-Security of Connected and Autonomous Vehicles (CAVs)". *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2022), pp. 6240–6259. DOI: 10.1109/TITS.2021.3085297.

[Tao+24] Ye Tao *et al.* "Zero-Knowledge Proof of Distinct Identity: a Standard-compatible Sybil-resistant Pseudonym Extension for C-ITS". *2024 IEEE Intelligent Vehicles Symposium (IV)*. 2024, pp. 1828–1835. DOI: 10.1109/IV55156.2024.10588511.

[Tar24] Tarlogic. *BSAM: Bluetooth Security Assessment Methodology*. 2024. URL: https://www.tarlogic.com/bsam/.

[TB19] Jan Trauernicht and Norbert Bißmeyer. "Deterministic Sybil attack exclusion in cooperative-intelligent transportation systems". *17th escar Europe : embedded security in cars (Konferenzveröffentlichung)*. Ruhr-Universität Bochum, Universitätsbibliothek, 2019, pp. 44–58. DOI: 10.13154/294-6655.

[Tho24] Brian Thorne. ***python-can***. June 2024. URL: https://github.com/hardbyte/python-can.

[Tin23] Ken Tindell. *CAN Injection: keyless car theft*. CANIS Automotive Labs, Apr. 2023. URL: https://kentindell.github.io/2023/04/03/can-injection/.

[UNE21a] UNECE. *UN Regulation No. 155 - Cyber security and cyber security management system*. Tech. rep. Mar. 2021. URL: https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security.

[UNE21b] UNECE. *UN Regulation No. 156 - Software update and software update management system*. Tech. rep. Mar. 2021. URL: https://unece.org/transport/documents/2021/03/standards/un-regulation-no-156-software-update-and-software-update.

[Uni18] European Union. *Regulation (EU) 2018/858 of the European Parliament and of the Council*. Tech. rep. May 2018. URL: http://data.europa.eu/eli/reg/2018/858/oj/eng.

[Val23] Peter Valdes-Dapena. *Some auto insurers are refusing to cover certain Hyundai and Kia models*. CNN Business, Jan. 2023. URL: https://edition.cnn.com/2023/01/27/business/progressive-state-farm-hyundai-kia/index.html.

[VGB12] Roel Verdult, Flavio D. Garcia, and Josep Balasch. "Gone in 360 Seconds: Hijacking with Hitag2". *21st USENIX Security Symposium (USENIX Security 12)*. Bellevue, WA: USENIX Association, Aug. 2012, pp. 237–252. ISBN: 978-1-931971-959. URL: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/verdult.

[VGE15] Roel Verdult, Flavio D. Garcia, and Baris Ege. "Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer". *Supplement to the Proceedings of 22nd USENIX Security Symposium (Supplement to USENIX Security 15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 703–718. ISBN: 978-1-931971-232. URL: https://www.usenix.org/system/files/sec15_supplement.pdf.

[VHG19] Eric Verheul, Christopher Hicks, and Flavio D. Garcia. "IFAL: Issue First Activate Later Certificates for V2X". *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2019, pp. 279–293. DOI: 10.1109/EuroSP.2019.00029.

[Vica] VicOne. *Driving Automotive Cybersecurity Forward*. VicOne. URL: https://vicone.com.

[Vicb] VicOne. *Pwn2Own Automotive*. VicOne. URL: https://vicone.com/pwn2own-automotive.

[Vil] Car Hacking Village. *Car Hacking Village*. URL: https://www.carhackingvillage.com.

[VM14] Chris Valasek and Charlie Miller. *Adventures in Automotive Networks and Control Units*. Tech. rep. IOActive, Oct. 2014. URL: https://www.ioactive.com/wp-content/uploads/pdfs/IOActive_Adventures_in_Automotive_Networks_and_Control_Units.pdf.

[VM18] Chris Valasek and Charlie Miller. *A Survey of Remote Automotive Attack Surfaces*. Tech. rep. IOActive, May 2018. URL: https://ioactive.com/wp-content/uploads/2018/05/IOActive_Remote_Attack_Surfaces.pdf.

[VVB18] Aram Verstegen, Roel Verdult, and Wouter Bokslag. "Hitag 2 Hell – Brutally Optimizing Guess-and-Determine Attacks". *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. Baltimore, MD: USENIX Association, Aug. 2018. URL: https://www.usenix.org/system/files/conference/woot18/woot18-paper-verstegen.pdf.

[Wan+20] Yan Wang *et al.* "Recent Development of Security Issues of Black Hole and Gray Hole Attacks in V2X Network". *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes: IEEE Press, 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294227.

[Wan+21] Yunpeng Wang *et al.* "A Systematic Risk Assessment Framework of Automotive Cybersecurity". *Automotive Innovation* 4.3 (Aug. 2021), pp. 253–261. ISSN: 2522-8765. DOI: 10.1007/s42154-021-00140-6.

[WCL20] Haohuang Wen, Qi Alfred Chen, and Zhiqiang Lin. "Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT". *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 949–965. ISBN: 978-1-939133-17-5. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/wen.

[WGP21] Lennert Wouters, Benedikt Gierlichs, and Bart Preneel. "My other car is your car: compromising the Tesla Model X keyless entry system". *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021.4 (Aug. 2021), pp. 149–172. DOI: 10.46586/tches.v2021.i4.149-172.

[WJL15] Samuel Woo, Hyo Jin Jo, and Dong Hoon Lee. "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN". *IEEE Transactions on Intelligent Transportation Systems* 16.2 (Apr. 2015), pp. 993–1006. ISSN: 1558-0016. DOI: 10.1109/TITS.2014.2351612.

[WLZ19] Juan Wang, Karim Lounis, and Mohammad Zulkernine. "CSKES: A Context-Based Secure Keyless Entry System". *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 1. July 2019, pp. 817–822. DOI: 10.1109/COMPSAC.2019.00120.

[Wou+19] Lennert Wouters *et al.* "Fast, furious and insecure: passive keyless entry and start systems in modern supercars". *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019.3 (May 2019), pp. 66–85. ISSN: 2569-2925. DOI: 10.13154/tches.v2019.i3.66-85.

[Wou+20] Lennert Wouters *et al.* "Dismantling DST80-based Immobiliser Systems". *IACR Transactions on Cryptographic Hardware and Embedded Systems* (Mar. 2020), pp. 99–127. ISSN: 2569-2925. DOI: 10.13154/tches.v2020.i2.99-127.

[Wu+20] Wufei Wu *et al.* "A Survey of Intrusion Detection for In-Vehicle Networks". *IEEE Transactions on Intelligent Transportation Systems* 21.3 (Mar. 2020), pp. 919–933. ISSN: 1558-0016. DOI: 10.1109/TITS.2019.2908074.

[XYG06] Bin Xiao, Bo Yu, and Chuanshan Gao. "Detection and localization of Sybil nodes in VANETs". *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks* (Oct. 2006). DOI: 10.1145/1160972.1160974.

[Yao+21] Yingying Yao *et al.* "LPC: A lightweight pseudonym changing scheme with robust forward and backward secrecy for V2X". *Ad Hoc Networks* 123 (2021), p. 102695. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2021.102695.

[YP19] Takahito Yoshizawa and Bart Preneel. "Survey of Security Aspect of V2X Standards and Related Issues". *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*. 2019, pp. 1–5. DOI: 10.1109/CSCN.2019.8931311.

[YP23] Takahito Yoshizawa and Bart Preneel. "Post-Quantum Impacts on V2X Certificates – Already at The End of The Road". *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*. 2023, pp. 1–6. DOI: 10.1109/VTC2023-Spring57618.2023.10199793.

[YP24] Takahito Yoshizawa and Bart Preneel. "Intersections are Not Good for Your Privacy". *2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2024, pp. 166–171. DOI: 10.1109/WiMob61911.2024.10770318.

[YZOB19] Clinton Young, Joseph Zambreno, Habeeb Olufowobi, and Gedare Bloom. "Survey of Automotive Controller Area Network Intrusion Detection Systems". *IEEE Design & Test* 36.6 (Dec. 2019), pp. 48–55. ISSN: 2168-2364. DOI: 10.1109/MDAT.2019.2899062.

[Zhu+24] Yaling Zhu *et al.* "Sybil Attacks Detection and Traceability Mechanism Based on Beacon Packets in Connected Automobile Vehicles". *Sensors* 24.7 (2024). ISSN: 1424-8220. DOI: 10.3390/s24072153.