On the Insecurity of Vehicles Against Protocol-Level Bluetooth Threats

Daniele Antonioli
EURECOM
Biot, France
daniele.antonioli@eurecom.fr

Mathias Payer
EPFL
Lausanne, Switzerland
mathias.payer@nebelwelt.net

***Abstract*—Cars are some of the most security-critical consumer devices. On the one hand, owners expect rich infotainment features, including audio, hands-free calls, contact management, or navigation through their connected mobile phone. On the other hand, the infotainment unit exposes exploitable wireless attack surfaces. This work evaluates protocol-level Bluetooth threats on vehicles, a critical but unexplored wireless attack surface. These threats are crucial because they are portable across vehicles, and they can achieve impactful goals, such as accessing sensitive data or even taking remote control of the vehicle. Their evaluation is novel as prior work focused on other wireless attack surfaces, notably Bluetooth implementation bugs. Among relevant protocol-level threats, we pick the KNOB and BIAS attacks because they provide the most effective strategy to impersonate arbitrary Bluetooth devices and are not yet evaluated against vehicles.**

Testing vehicles is challenging for several reasons, and we had to design a cost-effective methodology based on hybrid lab/on the road experiments. We evaluated 5 popular infotainment units (e.g., KIA and Toyota units) in the lab and 3 recent cars (e.g., Suzuki and Skoda cars) in a controlled on-the-road environment. We describe our methodology in detail to allow other researchers to reproduce and extend our results. Our Bluetooth protocol-level security evaluation uncovers worrisome facts about the state of vehicular security. For example, all tested devices are vulnerable to BIAS and KNOB, despite the patches in the Bluetooth standard. For example, the standard mandates keys with 7 bytes of entropy, but the tested devices accept keys with 1 byte of entropy. Moreover, all tested devices employ weak and outdated Bluetooth security parameters (e.g., weak authentication protocols and ciphers).

I. Introduction

Vehicles, such as cars, trucks, and motorbikes, employ numerous *wireless technologies*. For example, they use Bluetooth for hands-free services, cellular to access the Internet, Wi-Fi to share an Internet connection and keyless entry systems to wirelessly lock and unlock. It is estimated that since 2020 *one in five* vehicles employed wireless technologies, accounting for more than 25% of a billion cars [1].

Wireless connectivity exposes vehicles to *remote wireless attacks* [2] where an adversary in wireless range with the vehicle sends maliciously crafted packets to achieve impactful goals (e.g., data theft or remote code execution and control). One can classify wireless attacks based on the target wireless technology. In Section VI, we present a comprehensive discussion about those attacks covering, among others, Bluetooth, Wi-Fi, cellular, and keyless entry systems.

This work focuses on *protocol-based Bluetooth threats*, a new attack surface for vehicles. This attack surface covers threats exploiting the Bluetooth standard, such as attacks on Bluetooth pairing and session establishment. In contrast, prior work on automotive security focused on Bluetooth implementation issues [3], [4], [5], configurations lacking Bluetooth security [6], or security testing methodologies [7]. Hence, our paper fills a research gap in vehicular security, including automotive Bluetooth security.

Protocol-level threats on automotive Bluetooth are not only unexplored but also *relevant*. Vehicles include infotainment units that rely on Bluetooth to exchange data. By attacking those units, an adversary may access sensitive information about the driver, such as contact lists or text messages, along with the ability to send malicious commands to the unit itself. Using a protocol-level attack the adversary can impersonate a trusted smartphone to a vehicle infotainment unit over Bluetooth and get arbitrary read and write capabilities. Even worse, since the attack does not depend on the unit's hardware and software details (it exploits a logic Bluetooth bug), the adversary can easily port the attack to other units, even from different vendors.

For our protocol-level Bluetooth security evaluation on vehicles, we selected the *BIAS* [8] (CVE-2020-10135 [9]) and *KNOB* [10] (CVE-2019-9506 [11]) attacks. Their combination provides an effective and reliable attack vector to impersonate Bluetooth devices. Moreover, the authors in [10], [8] have *not* evaluated vehicles, so our work extends theirs to an important and widespread class of devices. In short, the attacks allow arbitrary device impersonation by targeting vulnerabilities in the entropy negotiation and authentication phases of Bluetooth session establishment without requiring to trigger a new paring session (see Section II-B for more technical details).

Since testing vehicles is challenging for third-party researchers, we had to develop a cost-effective methodology. We decided to use a *hybrid* strategy. Firstly we tested five popular infotainment units from KIA, Toyota, Mazda, Nissan and Subaru on a lab bench. During our experiments, we realized that only the KIA and Toyota units were fully functional, and we had to narrow our scope to these two. Then, we evaluated 3 actual cars that we own (i.e., a Suzuki IGNIS from 2021, a SKODA Fabia from 2020, and a SKODA Octavia from 2021) on the road. We explain our methodology in detail in Section IV.

Then we attacked the KIA and Toyota units and the Suzuki and SKODA cars with KNOB and BIAS. While performing

the attacks, we observed their Bluetooth security parameters. Our evaluation uncovers worrisome facts about the state of vehicular security against protocol-level Bluetooth threats. Despite mitigations mandated by the Bluetooth standard, all our tested devices are vulnerable to the BIAS and KNOB attacks. Hence, it is trivial to impersonate a trusted device to the vulnerable units and to all units reusing the same Bluetooth firmware. Moreover, while the devices' pairing capabilities are relatively strong (e.g., devices propose authenticated pairing requiring user interaction), the session capabilities are inadequate (e.g., devices propose vulnerable authentication protocols and ciphers). We describe our evaluation results in detail Section V.

We summarize our contributions as follows:

- We present the first security evaluation of protocol-level Bluetooth threats on vehicles, an unexplored attack surface entailing high-impact and portable attacks such as information disclosure and remote code execution ones.
- We describe our hybrid experimental methodology where we evaluate five popular Bluetooth infotainment units on a lab bench and three recent cars on the road. Specifically, we show how we reverse-engineered the power port layout to power up the units, and how we gathered technical information about the units and the cars using manual and automated techniques.
- We empirically demonstrate the insecurities of vehicles against protocol-level Bluetooth threats. For example, we successfully exploit all tested devices using BIAS and KNOB, despite the security patches in the Bluetooth standard. Moreover, we report that all tested devices use insecure Bluetooth security parameters such as weak authentication protocols and ciphers. As a result, it is trivial to spoof a trusted smartphone to an infotainment unit and port this attack to other units regardless of the software and hardware details of the target.

a) Disclosure and ethics: We responsibly disclosed a preliminary version of this paper to the *Automotive Information Sharing and Analysis Center (AutoISAC)* [12] in early 2021 and also the last version of the paper. They acknowledged our findings and circulated the paper among the affected vendors. All our experiments were conducted in a safe environment without threatening any third-party devices or vehicles.

II. Protocol-Level Bluetooth Attack on Vehicles

In this section, we introduce vehicular Bluetooth, we explain why the protocol-level Bluetooth attack surface is both relevant and unexplored, and we describe the KNOB and BIAS protocol-level Bluetooth attacks.

A. Vehicular Bluetooth Security

Bluetooth is the de-facto standard wireless technology for low-power wireless services. It is specified in an open standard, maintained by the Bluetooth special interest group (SIG), and its latest version is 5.3 [13]. Several vehicle manufacturers, including Toyota, Ford, and Hyundai, are part of the Bluetooth SIG [14]. The standard specifies two modes of transport: Bluetooth Classic and Bluetooth Low Energy. Currently, vehicles employ only *Bluetooth Classic* and in the rest of the paper we refer to it for simplicity as Bluetooth or BT.

TABLE I: Bluetooth profiles used by vehicles. Each profile provides a specific wireless service and is protected by the pairing and session establishment protocols specified in the Bluetooth standard.

Bluetooth profile	Acronym	Vehicle action
Advanced audio distribution	A2DP	Stream music from a source
Audio/Video remote control	AVRCP	Control music/video player
Hands-free	HFP	Manage calls
Message access	MAP	Read SMS
OBject EXchange	OBEX	Send/receive data
PAN Network Encapsulation	BNEP	Join Internet connection
Phone book access	PBA	Read contacts
Serial Port	SPP	Emulate a serial port
SIM access	SAP	Access a SIM card

Vehicles, through their infotainment units, provide several Bluetooth services that are known in the standard as Bluetooth *profiles*. In Table I, we list some of the most common. For example, the hands-free profile (HFP) manages phone calls from the infotainment display and the phone book access profile (PBA) pushes phone contacts to the infotainment unit.

The Bluetooth standard includes security mechanisms that adopters must implement to be specification-compliant. In particular, it provides a *pairing* protocol to negotiate a pairing key (PK) acting as a root of trust. Two devices are said to be paired after a successful run of the pairing protocol. Such protocol is based on an authenticated elliptic curve Diffie-Hellman (ECDH) and is known in the Bluetooth standard as Secure Simple Pairing (SSP).

Other than pairing, Bluetooth provides a *session establishment* protocol to authenticate two paired devices, negotiate a short-lived session key (SK) with variable entropy and use SK to encrypt the link. If the devices support Secure Connections (SC), they use mutual authentication and the AES-CCM cipher during a session. Otherwise, they employ unilateral authentication and the E_0 stream cipher, which are known to be vulnerable [8], [15]. Pairing and session establishment are started by the connection initiator, known as the BT central, towards the connection responder, known as the BT peripheral.

B. Bluetooth Protocol-Level Threats

A *Bluetooth protocol-level* threat has severe consequences on vehicles. For example, breaking the Bluetooth session establishment protocol allows an attacker to access all data exchanged by the vehicle and a trusted smartphone (e.g., contact lists, SMS, voice calls, and instant messages). Moreover, an adversary can impersonate a trusted smartphone, start a secure session with a vehicle and send arbitrary commands to its infotainment unit. Since the unit is connected to the vehicle's internal CAN bus, the attacker can even remotely control the

vehicle using Bluetooth packets by using the infotainment unit as a gateway to the CAN bus.

The vehicular Bluetooth protocol-level attack surface is *unexplored* despite its relevance. Prior work focuses on Bluetooth implementation bugs on infotainment units as their firmware are written in memory-unsafe languages (e.g., C/C++). Instead, we focus on logic bugs that derive from issues in the Bluetooth standard. For example, if the session establishment authentication protocol is the standard is vulnerable, all infotainment units are exploitable, and the exploit is portable across them. Hence, evaluating protocol-level Bluetooth threats on vehicles fills an important research gap.

C. KNOB and BIAS Attacks

This paper evaluates the *BIAS [8]* and *KNOB [10]* protocol-level threats on automotive Bluetooth. Those attacks target authentication and entropy negotiation vulnerabilities in Bluetooth's *session establishment*. By chaining the attacks, an adversary can impersonate a trusted device and establish a secure session with a victim without having to repair with the victim device. For example, an attacker can approach any vehicle pretending to be a trusted smartphone and connect with that vehicle while impersonating the smartphone. Alternatively, she can approach a smartphone and impersonate a trusted vehicle. As a result of the attacks, the adversary can, among others, access sensitive information (e.g., contacts and text messages) and send arbitrary Bluetooth commands.

Before illustrating BIAS and KNOB, we describe a legitimate vehicular pairing process. Firstly, the driver pairs her smartphone with her car by searching her smartphone from the car infotainment screen. While pairing, the driver is typically required to confirm that the smartphone and the infotainment module display the same numeric code. This is called *Numeric Comparison association* and it should protect from Attacker-in-the-Middle (AitM) threats [13, p. 743]. Once the devices are paired they share a long-term PK and can establish secure sessions *without* requiring user interaction. The session establishment protocol has the following five phases:

1) Identification and connection
2) PK authentication (vulnerable to BIAS)
3) SK entropy negotiation (vulnerable to KNOB)
4) SK derivation from PK, nonces, and negotiated entropy
5) Start an encrypted session using SK

During a secure session, the vehicle and the smartphone can use one or more Bluetooth profiles, including the ones from Table I. Some of the profiles might trigger a one-time permission request from the smartphone during pairing. For example, on Android, if the user wants to transfer her contact list from the smartphone to the infotainment unit, then she is asked once to authorize the operation.

The KNOB and BIAS attacks can be chained to impersonate a Bluetooth device during session establishment without knowing the current SK or the long-term PK. We now present a technical description of those impersonation attacks. Without loss of generality, we assume that the smartphone is the connection initiator and the car (infotainment unit) is the responder.

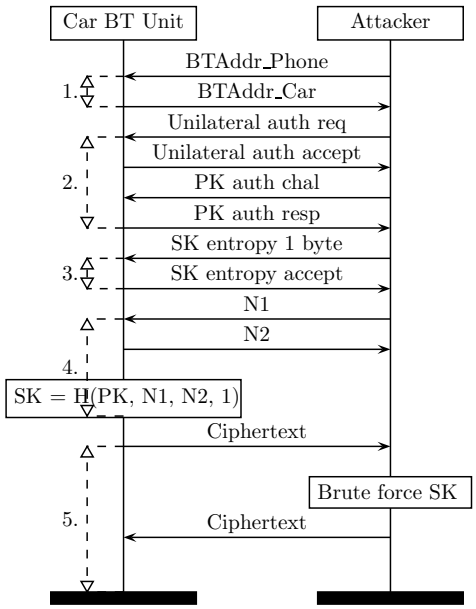

Fig. 1: Spoofing a trusted phone to a car infotainment unit via BIAS and KNOB attacks.

a) BIAS+KNOB phone impersonation: Figure 1 describes an attacker impersonating a trusted phone to a car via KNOB and BIAS. In 1. the adversary initiates the connection using the Bluetooth address of the phone; in 2. she performs a BIAS attack by negotiating a unilateral PK authentication protocol where the authenticator is always the connection initiator, asking the smartphone to authenticate with a challenge-response protocol, and completing PK authentication *without having to authenticate* (i.e., without proving possession of PK to the smartphone); in 3. the attacker exploits the KNOB attack by negotiating the lowest (yet specification-compliant) entropy value for SK. Such value is as low as 1 byte; in 4. the attacker and the smartphone derive a weak SK from PK, two nonces that they exchange, and the negotiated entropy value; in 5. the attacker waits for an encrypted packet from the smartphone, uses the ciphertext to brute force SK, and continues the session impersonating the car.

b) BIAS+KNOB infotainment unit impersonation: Figure 2 presents an attacker impersonating a trusted car infotainment unit to a smartphone via KNOB and BIAS. This attack differs from the previous one during phases 1. and 2. In particular, during 1. the attacker advertises her presence using the car Bluetooth address, and in 2. the attacker has to perform an extra trick to avoid authentication. Specifically, the attacker asks the victim to switch roles before authentication occurs, becomes the verifier, and does not have to authenticate to the victim. Then the attacker completes 3. 4. and 5. as in the phone impersonation attack.

c) KNOB and BIAS patches: The KNOB and the BIAS attacks have been disclosed in 2019 (CVE-2019-9506) and 2020 (CVE-2020-10135). The KNOB attack was partially addressed

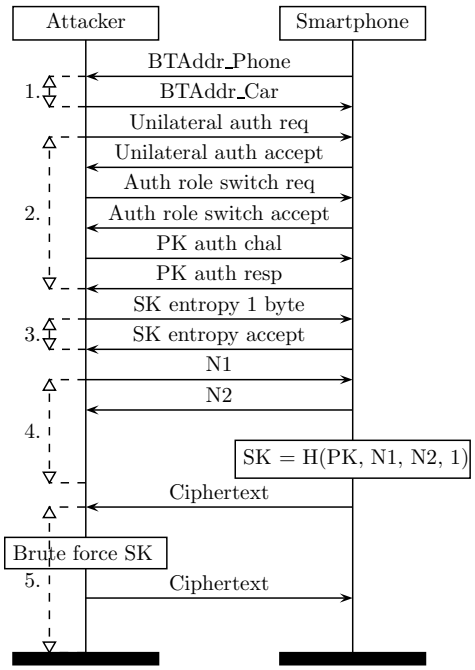

Fig. 2: Spoofing a trusted car infotainment to a smartphone via BIAS and KNOB attacks.

with an amendment to the Bluetooth standard forbidding encryption keys with entropy lower than 7 byte [16]. As a result of this amendment, popular IT vendors such as Google, Microsoft, and Apple shipped security patches. However, it is *not* clear if vehicles are patched against KNOB. After the BIAS attacks report, the Bluetooth standard was amended again with generic recommendations, such as avoiding switching roles during session authentication [17]. However, it is unclear if IT and vehicles vendors patched their devices against BIAS.

III. Threat Model

In this section, we describe our system and attacker models.

A. System model

We consider a system with a vehicle equipped with a Bluetooth infotainment unit (e.g., a car or a truck), a smartphone, and a user. The user already paired the vehicle infotainment unit with her smartphone using the strongest Bluetooth security mode available (e.g., Bluetooth Secure Simple Pairing, Numeric Comparison association, and Secure Connections). Moreover, the user already accepted the one-time permissions (e.g., allow contact access and audio services).

As a consequence of pairing, the infotainment module and the smartphone can establish secure connections using Bluetooth's session establishment protocol. A secure connection can be started automatically when the vehicle is in Bluetooth range with the smartphone or manually by the user via the vehicle's user interface (UI) or the smartphone's UI.

B. Attacker Model

Our attacker model considers a *wireless* attacker targeting a secure Bluetooth connection between a vehicle infotainment unit and a smartphone using protocol-level attacks. A protocol-level attack exploits vulnerabilities in the Bluetooth standard and can be repurposed to any infotainment unit. Specifically, the attacker targets the Bluetooth session establishment protocol chaining the KNOB and BIAS attacks presented in Section II-C.

The attacker has two goals. (1) impersonating a trusted smartphone to an infotainment unit; (2) AitM a secure connection between a smartphone and an infotainment unit. These goals entail severe consequences for the victim. For example, the attacker can access all services and (sensitive) data or inject malicious packets, including rogue commands and remote code execution (RCE) payloads.

The adversary has limited knowledge about the victims and the usual capabilities of a wireless attacker. She knows their Bluetooth addresses and the other Bluetooth information transmitted in cleartext, such as Bluetooth names and supported profiles. She can eavesdrop (encrypted) Bluetooth packets, jam the Bluetooth spectrum, and craft custom packets. However, she does not have physical access to the target devices (e.g., she cannot install custom software on them), does not know their Bluetooth pairing and session keys (i.e., PK and SK), and did not observe their secure pairing procedure.

IV. Methodology

We now describe our methodology consisting of experiments in the lab and on the road.

A. Rationale

a) Possible approaches: Testing Bluetooth infotainment units is challenging for third-party researchers. To get the best possible setup (e.g., access to vendor-specific tools, pre-production units, and official documentation) a researcher has to approach a vendor and sign a non-disclosure agreement (NDA). Another strategy is testing units *in isolation* on a lab bench. This approach minimizes safety risks and economic costs, however, it is time-consuming (e.g., selection of the relevant units, lab setup, etc.) [18]. A third evaluation strategy is testing owned cars on the road. This approach is the most realistic (e.g., test the interaction of the unit with a CAN bus), but it is also the most risky and costly. We exclude testing rented cars from our scope due to ethical and legal concerns.

b) Our approach: We eventually decided to use a *hybrid* approach combining experiments in the lab and on the road. We excluded the NDA-based approach as it prevents publishing results and does not scale with different (and competing) vendors. We designed our experiments in two sets. Firstly, we tested 5 Bluetooth infotainment modules on a lab bench. We selected these based on popularity with the help of Privacy4Cars [19], a company specialized in automotive privacy analyses. Eventually, we tested units used by KIA, Toyota, Mazda, Nissan, and Subaru that we bought on eBay. In the second experimental set, we evaluated 3 actual cars that we own in a controlled environment (i.e., Suzuki IGNIS, Skoda Fabia, and Skoda Octavia).

Fig. 3: Two of the five Bluetooth infotainment modules that we tested: a KIA 96560-B2211CA unit (on the left) and a Toyota PT546-00170 unit (on the right).

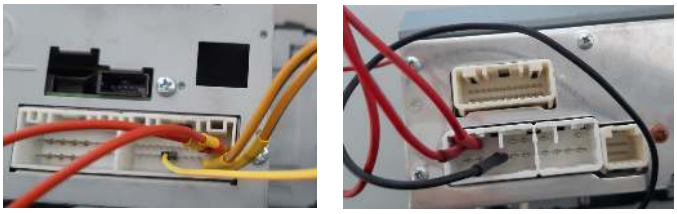

Fig. 4: Powering up Bluetooth infotainment units. On the left, we show how to power up the KIA 96560-B2211CA module by connecting five wires to its 24 pin power port. On the right, we display how to power up the Toyota PT546-00170 unit by connecting three wires to its 16 pin power port.

B. Lab Methodology

For our lab experiments we bought five popular infotainment units from eBay costing us approximately 3,000 EUR. Here we list their advertised name:

1) KIA 96560-B2211CA (left unit in Figure 3)
2) Toyota PT546-00170 (right unit in Figure 3)
3) Mazda NA1P611J0
4) Nissan 253914HB0A
5) Subaru 86271SG630

a) Powering up the units: First, we had to *power up* the units, which can be challenging for several reasons. There is *no standard power port* for infotainment units and getting information about the power port is tricky unless you pay a fee to the vendor. Even worse, two infotainment units might have the same power port shape but a distinct pin layout. Moreover, an infotainment unit might require a vendor-specific electric signal (e.g., a CAN signal) sent in conjunction with the power signals to switch on the unit.

We attempted to power up the five units by manually determining the location of the power port, reverse-engineering the pin layout, and trying different signal combinations. We used a power supply with tunable voltage and amperage, jumper cables, a breadboard, and discrete components. To get information on various power port layouts, we consulted https://pinouts.ru/.

From our power-up tests, we realized that the Mazda and Nissan units were *not testable*. We did not attempt to power up the Mazda unit as we found that it lacked the Bluetooth subsystem despite being advertised as a functional unit. Instead, the Nissan module does not power up at all, most probably due to a broken power circuit.

TABLE II: Relevant technical specifications of the KIA and Toyota Bluetooth infotainment units. The Used By column indicates the vehicles that we know are employing such unit and is a lower bound as many more vehicles are employing them. CSR stands for Cambridge Silicon Radio. We use n/a if a piece of information is not available. We redact Bluetooth addresses for privacy reasons.

	KIA 96560-B2211CA	**Toyota PT546-00170**
Manuf.	Hyundai (Mobis)	Toyota
Year	2014	2012
Used by	KIA Soul (2015, 2016)	Toyota 86 (2017, 2018, 2019), Toyota Corolla (2017, 2018)
BT Manuf.	Hyundai	Pioneer
BT Vers.	3.0	3.0
BT ID	B021345	D049474
BT Design	Ax2xxB1xx	32948, 42100
BT Host	n/a	Qualcomm (QTIL), v4.0+HS
BT Ctrl	n/a	Alps Alpine, v3.0
BT Firmw.	CSR 8241	CSR 9079
BT Addr.	Redacted	Redacted
BT Name	KIA MOTORS	My Toyota
BT Class	0x240408	0x340408
BT Profile	A2DP, AVRCP, HFP	SPP, OBEX, A2DP, AVRCP, HFP, MAP
FCC ID	AVC31B2AN	AJDK068
FCC Test	2014	2013
Wi-Fi	Yes	No

We managed to power up the KIA, Toyota, and Subaru units. In Figure 4, we show the KIA unit power-up setup on the left and the Toyota power-up setup on the right. The KIA module has 24 pins and powers up by connecting five wires. Pins 11, 12, and 18 expect a positive voltage, while pins 23, and 24 connect to the ground. The Toyota unit has a 12 pin power port and requires three wires. Pin 4 and 3 expect positive voltage, while pin 7 connect to the ground.

b) Check Bluetooth connectivity: Before embarking on time-consuming static and dynamic analyses, it is fundamental to check if the infotainment Bluetooth subsystem behaves as expected. In our experiments, we checked the Bluetooth connectivity of the KIA, Toyota, and Subaru units. In particular, we tried to pair the units with a phone and perform basic interactions (e.g., accept permissions, connect, and stream music). This phase was helpful to identify a problem with the Subaru module. In particular, the module could complete the pairing process with our phone, but then it stopped working. The user interface was still responsive, but the Bluetooth firmware seemed corrupted. Since the firmware cannot be gathered and re-flashed without collaborating with the vendor, we had to discard the Subaru module. The KIA and Toyota modules were working as expected instead, leaving us to experiment with 2 out of 5 modules.

c) Finding units' technical specification: In the third phase of our lab experiments, we gathered more technical information about the KIA and Toyota units. This was not

an easy task as there is no public and central repository for infotainment units but the information is partial, scattered across the Internet, and sometimes behind a paywall.

In Table II, we summarize what we found about the KIA and Toyota modules. The first three rows are information about the unit manufacturer, year of production, and usage. We note that the usage row is just a lower bound, as different car models and brands might use the *same* infotainment unit. For example, the 96560-B2211CA unit was advertised as a KIA unit, but is produced by Hyundai and might be used in Hyundai cars.

The other rows in Table II provide technical details about Bluetooth (and Wi-Fi). Such data was collected using manual and automated techniques. In particular, we noted down the serial numbers printed on the units, and we googled them to acquire more information. Two useful serial numbers are the Federal Communications Commission identifier (FCC ID) that can be queried via https://fccid.io/, and the Bluetooth ID that can be queried via https://launchstudio.bluetooth.com/Listings/Search. Moreover, we used standard Bluetooth Linux tools, such as `hcitool` and `sdptool`, to automatically interact with the units over-the-air. For example, to get the list of supported Bluetooth profiles (BT Profile row in Table II) we used the `sdptool browse BTADD` command where `BTADD` is the Bluetooth address of the target device.

d) Attacking the units: In the last experimental phase, we attacked the KIA and Toyota units using KNOB and BIAS. See Section V for the details.

C. On The Road Methodology

For our on the road experiments we tested the following cars that we own on the road in a controlled setup:

1) Suzuki IGNIS from 2021
2) Skoda Fabia from 2020
3) Skoda Octavia 3 from 2021

a) Finding cars' technical specification: Firstly, we gathered as much information as possible about the target cars. As for the lab experiments, we used manual and automated techniques. For example, we manually inspected all the car infotainment menus. In Figure 5 we show a picture of the Fabia system information menu including, among others, serial numbers about Bluetooth (C190), hardware (690), and software (8740). We also automatically queried the cars using Bluetooth command line tools to gather various information such as Bluetooth name, version, device class, firmware versions, list of supported profiles, and others. In Table III, we summarize the technical information

b) Comparing technical specifications: We used Table III to compare the analyzed cars and extract interesting facts about their Bluetooth versions, manufacturers, and firmware. We now summarize the most important findings.

We expected to find recent Bluetooth versions and modern cars, but this is not the case. The cars ship with Bluetooth 3.0 (2009) and 4.1 (2013), despite being the models from 2020 and 2021. Moreover, if we focus on SKODA, the Octavia supports an older Bluetooth version than the Fabia, despite being newer. These are strong indicators that even if a car is new, it might ship with old and vulnerable Bluetooth versions and unpatched code.

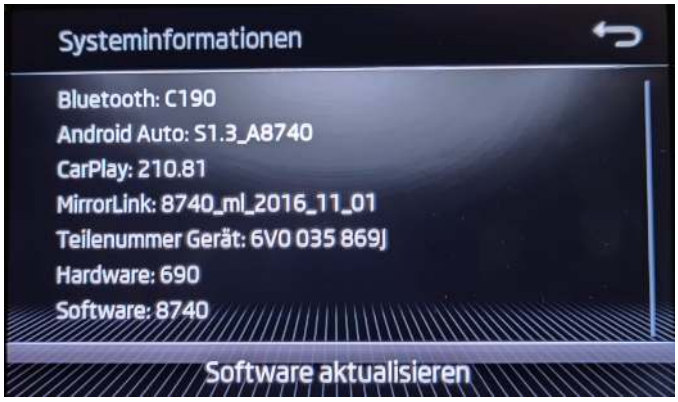

Fig. 5: Skoda Fabia infotainment unit's system menu.

TABLE III: Technical specification of the Suzuki IGNIS and Skoda Fabia and Octavia that we tested. If information is unavailable, we indicate with n/a. We redact Bluetooth addresses (BT Addr.) for privacy reasons.

	Suzuki IGNIS	**Skoda Fabia**	**Skoda Octavia**
Year	2021	2020	2021
BT Manuf.	Harman	Toshiba	Harman
BT Vers.	3.0	4.1	3.0
BT ID	n/a	n/a	n/a
BT Firmw.	CSR 8241	Toshiba 3328	CSR 8241
BT Addr.	Redacted	Redacted	Redacted
BT Name	Suzuki	Skoda BT 1684	Skoda BT
BT Class	0x360408	0x360408	0x360408
BT Profile	SPP, A2DP, AVRCP, HFP, PBA	A2DP, AVRCP, HFP	SPP, MNS, HFM, PBAP, AVRCP, A2DP
Wi-Fi	No	No	No

We also confirm that the same car manufacturer can employ units from different and competing brands. For example, the Skoda Fabia ships with a Toshiba subsystem, while the Octavia with a Harman one. As a result, even if we compare two cars from the same manufacturer, we can expect a different level of security because of heterogeneous infotainment units.

Another interesting finding is that the IGNIS and Octavia share the *same* Bluetooth setup (i.e., Harman manufacturer and CSR 8241 firmware). This fact experimentally confirms that infotainment units are shared across vendors. Hence, a protocol-level Bluetooth threat that is effective on the IGNIS can be ported to any other infotainment unit using Harman with CSR 8241, including the tested Octavia.

c) Attacking the cars: In the second experimental phase, we attacked the cars' infotainment units with KNOB and BIAS. The attacks were performed on the road in a controlled environment and did not involve third-party cars (see section V for the details).

V. Evaluation

In this section we report our evaluation setup and results.

A. Setup

a) Attack scenario: Our evaluation targeted two infotainment units attacked in the lab (i.e., KIA 96560-B2211CA and Toyota PT546-00170) and three cars exploited on the road (i.e., Suzuki IGNIS, Skoda Fabia, and Skoda Octavia). We attacked them with a smartphone impersonation attack exploiting KNOB and BIAS. The attack is presented in Section II-B (see Figure 1). We conducted the smartphone impersonation attack as follows. We paired a (benign) smartphone with the target infotainment unit and granted all the required permissions. Then, we started a secure session with the victim unit as a trusted smartphone, and we tried to bypass the pairing key authentication phase (BIAS) and to negotiate a session key with one byte of entropy (KNOB).

b) Attack device: Our attack device consists of a Cypress development board (i.e., CYW920819) connected via USB to a Linux laptop (i.e., Thinkpad X1). This is the same attack device used by the BIAS and KNOB developers and documented at https://github.com/francozappa/bias. The Cypress board acts as the Bluetooth controller and can be patched at runtime using a dynamic Bluetooth patching framework called internalblue [20]. The laptop plays the role of the Bluetooth host and can send (vendor-specific) commands to the board. We also patched the laptop's Linux kernel to enable the reception of diagnostic messages from the board, including link-layer packets. Overall, such an attack device is low-cost, based on open-source software, and easy to reproduce.

B. Results

We now summarize the evaluation results about the five tested Bluetooth infotainment units with the help of Table IV.

a) Pairing capabilities: As shown in the first section of Table IV, the devices support *relatively strong* Bluetooth pairing configurations. All devices employ SSP (i.e., ECDH key agreement), and declare input-output capabilities (i.e., display with Yes/No buttons). Hence, during pairing, they use Numeric Comparison association, and the user has to confirm that she sees the same numeric code on the infotainment unit and the smartphone. Notably, the Toyota unit does not declare AitM protection despite providing input-output capabilities.

b) Session capabilities: As we show in the second section of Table IV, the units support *weak* Bluetooth secure session capabilities. None of the devices support SC, and this fact has two consequences. Firstly, the devices default to the weak and unilateral authentication protocol. Secondly, the devices employ a legacy stream cipher (E_0), providing no integrity protection. Overall, these capabilities facilitate the exploitation of KNOB and BIAS as the attacker does not have to downgrade the authentication protocol and has to brute force a weak cipher.

c) Session issues: As depicted in the third section of Table IV, *all* the infotainment units that we evaluated are vulnerable to device impersonation via KNOB and BIAS. In particular, despite the KNOB attack being patched in 2019, the devices still accept negotiating session keys with 1 byte of entropy while the standard mandates a minimum of 7 bytes [16]. Moreover, all devices are vulnerable to the role switch authentication trick demonstrated in the BIAS paper, despite the role switch mitigations in the standard [17]. These facts experimentally confirm how brittle vehicular security is against protocol-level Bluetooth threats.

d) Pairing issues: We report extra threats related to pairing (see last section of Table IV). The Skoda cars are *always discoverable*, meaning that an attacker can easily find their Bluetooth addresses. Discoverability can be manually disabled via a dedicated menu. Even worse, the Skoda cars and the KIA module are *always pairable*. Hence, an attacker who knows their Bluetooth address can pair with them anytime. All devices except the Suzuki IGNIS car are vulnerable to *pairing association downgrade attacks*. In this attack the adversary pairs with a victim (while spoofing a trusted device) and pretends to support no I/O capabilities to trigger Just Works which is not authenticated and does not mandate user interaction. Hence, despite the relatively strong pairing capabilities, the devices are also vulnerable to protocol-level attacks during pairing.

VI. Related Work

In this section we present the related work divided by topic.

a) Remote wireless attacks on vehicles: Several research papers covered wireless attacks on vehicles, but none of them looked into protocol-level Bluetooth vulnerabilities so far. For example, in [21] the authors demonstrated how to track vehicles using wireless tire pressure monitoring systems. Keyless entry systems received quite a lot of attention. In [22], the authors demonstrated a practical key recovery attack against KeeLoq, a block cipher used to authenticate keyless systems. KeeLoq was also found vulnerable to differential power analysis side channel attacks [23]. Relay attacks were found to be very effective on different keyless entry systems [24]. Wireless immobilizers, a component of keyless entry systems, were also attacked. For example, in [25] researchers demonstrated three practical wireless key recovery attacks on the proprietary stream cipher used by Hitag2, the most popular car immobilizer in 2012. The same team in [26] presented similar key recovery attacks on Megamos Crypto, another popular immobilizer. Hitag2 security was also revisited in 2017 [27]. It appears that even modern keyless entry systems are still insecure as demonstrated by [28], where the authors were able to clone the key fob of a Tesla Model S in seconds by exploiting various weaknesses of the Tesla proprietary keyless system.

b) Bluetooth protocol-level vulnerabilities: Several papers were published about protocol-level Bluetooth issues but none of them evaluated those issues on vehicles. It is known that Bluetooth secure simple pairing vulnerable to AitM and reflection attacks [29], [30], [31]. An attacker can also perform an invalid curve attack on Bluetooth pairing as the pairing protocol is based on elliptic curve cryptography and not all devices check the validity of the remote public keys [32]. Bluetooth association was also found vulnerable to downgrade and method confusion attacks [33], [34]. Pairing downgrade

TABLE IV: Evaluation results. Devices have relatively strong pairing capabilities resulting in authenticated pairing with Numeric Comparison. The session capabilities are weak as the devices employ unilateral authentication and the E_0 legacy cipher. As all devices are vulnerable to KNOB and BIAS, it is trivial to spoof a trusted smartphone to an infotainment unit, despite the mitigations mandated by the Bluetooth standard [16], [17]. For example, the tested devices are still vulnerable to 1-byte entropy downgrades and role-switching to bypass authentication. Moreover, three out of five devices are always pairable and vulnerable to Just Works downgrade attacks.

	KIA 96560-B2211CA	**Toyota PT546-00170**	**Suzuki IGNIS**	**Skoda Fabia**	**Skoda Octavia**
	Car unit	Car unit	Car	Car	Car
Pairing capabilities					
Secure Simple Pairing (SSP)	Yes	Yes	Yes	Yes	Yes
Input Output	Display	Display	Display	Display	Display
Authentication Requirement	AitM	None	AitM	AitM	AitM
Association	Num Comp	Num Comp	Num Comp	Num Comp	Num Comp
Session capabilities					
Secure Connections (SC)	No	No	No	No	No
Unilateral authentication	Yes	Yes	Yes	Yes	Yes
E_0 cipher (weak)	Yes	Yes	Yes	Yes	Yes
Session issues					
Entropy downgrade	1 byte	1 byte	1 byte	1 byte	1 byte
Role switch auth bypass	Yes	Yes	Yes	Yes	Yes
Vulnerable to KNOB & BIAS	Yes	Yes	Yes	Yes	Yes
Pairing issues					
Always Discoverable	No	No	No	Yes	Yes
Always Pairable	Yes	No	No	Yes	Yes
Just Works Downgrade	Yes	Yes	No	Yes	Yes

using Just Works was employed in the BLURtooth cross-transport attacks [35]. Prior version of Bluetooth pairing, known as legacy pairing, were also found vulnerable [36], [37], [38], [39], [40].

c) CAN bus security: Most of the automotive security research work has been focusing on the CAN bus as it was designed with no security requirements. In [41], the authors demonstrated that an attacker who has access to the CAN bus can disable inputs from the driver, circumvent safety-critical systems, and control the engine, the brakes, the radio, and the lock. Similar attacks were also proven effective on trucks [42]. In [43] the authors evaluated how to fingerprint a driver based on CAN bus data taken from his car. DoS attacks on the CAN bus were also evaluated [44]. Several CAN mitigations were proposed over the years, including a legacy-compliant authentication scheme based on hash-based message authentication codes (HMAC) [45], and an authentication scheme that protects from remote and local attackers with code execution capabilities on the vehicle network [46]. Several CAN intrusion detection systems based on clock signals [47], CAN network entropy [48], time intervals [49], voltage levels [50] and others [51] were also proposed. Researcher also developed tools to automate the security evaluation of the CAN bus. For example, CANvas [52] is an automotive network mapping tool for the CAN bus to map ECUs and READ [53], LibreCAN [54], and AutoCAN [55] were developed to automate reverse-engineering of proprietary CAN messages.

d) USB infotainment protocols: Alternatively to Bluetooth, there are standardized and proprietary protocols to connect a smart device to an infotainment unit over USB. These technologies are less user-friendly as, unlike Bluetooth, they require cables. The two most popular ones are Android Auto [56] developed by Google and iOS CarPlay [57] provided by Apple. MirrorLink used to be popular and was evaluated in [58].

VII. Conclusion

Bluetooth provides a vast attack surface for modern vehicles, including cars and trucks. This work presents the first security evaluation of protocol-level Bluetooth threats on vehicles, an unexplored but relevant attack surface. Prior work focused on Bluetooth implementation bugs (e.g., memory corruptions) or insecure Bluetooth setups (e.g., no pairing). Instead, we focus on bugs affecting the Bluetooth standard (i.e., vulnerabilities in Bluetooth pairing and session establishment security protocols). Protocol-level attacks enable reaching impactful goals, such

as disclosing sensitive data and injecting malicious commands. They are also easy to port across units as they do not depend on hardware and software configurations.

We describe a hybrid methodology to assess Bluetooth protocol-level threats on vehicles. We suggest starting with experiments in the lab to minimize safety risks and economic costs. Then proceed testing actual cars on the road in a controlled environment. We detail each step that we used in the lab and on the road (e.g., reversing power pin layouts and manual and automatic technical specification gathering). Our methodology enabled us to extract valuable facts about the five infotainment units tested in the lab and the three cars evaluated on the road. For example, new cars might ship with outdated infotainment systems containing legacy code and insecure Bluetooth versions, or cars from the same brand can employ infotainment units from different companies.

We empirically uncover several worrisome facts about the state of vehicular security against protocol-level Bluetooth threats. For example, all tested devices are vulnerable to KNOB and BIAS protocol-level attacks, despite the Bluetooth standard mandates to mitigate them. The attacks enable spoofing a trusted smartphone to an infotainment unit to access sensitive data or send malicious commands to the unit. Moreover, the tested devices are vulnerable to other protocol-level threats related to paring (e.g., Just Works downgrade attacks). This work scratched the protocol-level Bluetooth attack surface, and we hope to kick-start more papers on this subject.

ACKNOWLEDGMENT

We would like to acknowledge Andrea Amico from Privacy4Cars [19] for funding and industrial expertise and Jean-Micheal Crepel for helping with the laboratory setup, reverse engineering, and experiments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 850868), DARPA HR001119S0089-AMP-FP-034, and ONR award N00014-18-1-2674. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our sponsors.

REFERENCES

[1] GSMA, "Emerging car trends and market predictions," https://www.gsma.com/iot/wp-content/uploads/2016/09/Connected-car-emerging-trends-and-market-predictions.pdf, 2016.

[2] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *black hat USA*, vol. 2014, p. 94, 2014.

[3] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*, vol. 4. San Francisco, 2011, pp. 447–462.

[4] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *Def Con*, vol. 21, no. 260-264, pp. 15–31, 2013.

[5] T. K. S. Lab, "Experimental Security Assessment on Lexus Cars," https://tinyurl.com/2p9fnrbx, 2020.

[6] Privacy4Cars, "Carsblues vehicle hack exploits vehicle infotainment systems allowing access to call logs, text messages and more," https://privacy4cars.com/data-in-cars/responsible-disclosure-and-p4c-bug-bounty/#carblues, 2018.

[7] M. Cheah, S. A. Shaikh, O. Haas, and A. Ruddle, "Towards a systematic security evaluation of the automotive Bluetooth interface," *Vehicular Communications*, vol. 9, pp. 8–18, 2017.

[8] D. Antonioli, N. O. Tippenhauer, and K. Rasmussen, "BIAS: Bluetooth Impersonation AttackS," in *Proceedings of Symposium on Security and Privacy (S&P)*. IEEE, May 2020.

[9] N. NVD, "Cve-2020-10135 detail," https://nvd.nist.gov/vuln/detail/CVE-2020-10135, 2020.

[10] D. Antonioli, N. O. Tippenhauer, and K. Rasmussen, "The KNOB is broken: Exploiting low entropy in the encryption key negotiation of Bluetooth BR/EDR," in *Proceedings of the USENIX Security Symposium*. USENIX, August 2019.

[11] N. NVD, "Cve-2019-9506 detail," https://nvd.nist.gov/vuln/detail/CVE-2019-9506, 2019.

[12] Automotive Information Sharing and Analysis Center AUTO-ISAC, "AUTO-ISAC Homepage," https://automotiveisac.com/, 2021.

[13] Bluetooth SIG, "Bluetooth Core Specification v5.3," https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=521059, 2021.

[14] ——, "Become a Bluetooth SIG Member," https://www.bluetooth.com/develop-with-bluetooth/join/, 2021.

[15] J. Padgette, K. Scarfone, and L. Chen, "Guide to Bluetooth Security," *NIST special publication*, vol. 800, p. 121, 2017.

[16] Bluetooth SIG, "Key Negotiation of Bluetooth," https://www.bluetooth.com/learn-about-bluetooth/key-attributes/bluetooth-security/statement-key-negotiation-of-bluetooth/, 2019.

[17] ——, "Bluetooth SIG Statement Regarding the Bluetooth Impersonation Attacks (BIAS) Security Vulnerability," https://www.bluetooth.com/learn-about-bluetooth/key-attributes/bluetooth-security/bias-vulnerability/, 2020.

[18] C. Miller and C. Valasek, "Car hacking: for poories," Tech. rep., IOActive Report, Tech. Rep., 2015.

[19] Privacy4Cars, "Privacy4Cars Homepage," https://privacy4cars.com/, 2022.

[20] D. Mantz, J. Classen, M. Schulz, and M. Hollick, "InternalBlue Bluetooth binary patching and experimentation framework," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 2019, pp. 79–90.

[21] I. Rouf, R. D. Miller, H. A. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study." in *USENIX Security Symposium*, vol. 10, 2010.

[22] S. Indesteege, N. Keller, O. Dunkelman, E. Biham, and B. Preneel, "A practical attack on KeeLoq," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2008, pp. 1–18.

[23] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme," in *Annual International Cryptology Conference*. Springer, 2008, pp. 203–220.

[24] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2011.

[25] R. Verdult, F. D. Garcia, and J. Balasch, "Gone in 360 seconds: Hijacking with hitag2," in *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, 2012, pp. 237–252.

[26] R. Verdult, F. D. Garcia, and B. Ege, "Dismantling megamos crypto: Wirelessly lockpicking a vehicle immobilizer," in *Supplement to the Proceedings of 22nd {USENIX} Security Symposium (Supplement to {USENIX} Security 13)*, 2013, pp. 703–718.

[27] R. Benadjila, M. Renard, J. Lopes-Esteves, and C. Kasmi, "One car, two frames: attacks on hitag-2 remote keyless entry systems revisited," in *11th USENIX Workshop on Offensive Technologies WOOT 17*, 2017.

[28] L. Wouters, E. Marin, T. Ashur, B. Gierlichs, and B. Preneel, "Fast, furious and insecure: Passive keyless entry and start systems in modern supercars," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 66–85, 2019.

[29] K. Haataja and P. Toivanen, "Two practical man-in-the-middle attacks on Bluetooth secure simple pairing and countermeasures," *Transactions on Wireless Communications*, vol. 9, no. 1, pp. 384–392, 2010.

[30] D.-Z. Sun, Y. Mu, and W. Susilo, "Man-in-the-middle attacks on Secure Simple Pairing in Bluetooth standard v5. 0 and its countermeasure," *Personal and Ubiquitous Computing*, vol. 22, no. 1, pp. 55–67, 2018.

[31] T. Claverie and J. L. Esteves, "BlueMirror: Reflections on Bluetooth Pairing and Provisioning Protocols," in *Proceedings of Workshop on offensive security (WOOT)*, 2021.

[32] E. Biham and L. Neumann, "Breaking the bluetooth pairing–fixed coordinate invalid curve attack," http://www.cs.technion.ac.il/~biham/BT/bt-fixed-coordinate-invalid-curve-attack.pdf, 2018.

[33] K. Hypponen and K. M. Haataja, "Nino man-in-the-middle attack on Bluetooth Secure Simple Pairing," in *Proceedings of the International Conference in Central Asia on Internet*. IEEE, 2007, pp. 1–5.

[34] M. von Tschirschnitz, L. Peuckert, F. Franzen, and J. Grossklags, "Method Confusion Attack on Bluetooth Pairing," in *Proceedings of Symposium on Security and Privacy (S&P)*. IEEE, 2021.

[35] D. Antonioli, N. O. Tippenhauer, K. Rasmussen, and M. Payer, "BLURtooth: Exploiting Cross-Transport Key Derivation in Bluetooth Classic and Bluetooth Low Energy," in *Proceedings of the Asia conference on Computer and Communications Cecurity (ASIACCS)*. ACM, May 2022.

[36] M. Jakobsson and S. Wetzel, "Security weaknesses in Bluetooth," in *Proceedings of the Cryptographers' Track at the RSA Conference*. Springer, 2001, pp. 176–191.

[37] A. Levi, E. Çetintaş, M. Aydos, Ç. K. Koç, and M. U. Çağlayan, "Relay attacks on Bluetooth authentication and solutions," in *Proceedings International Symposium on Computer and Information Sciences*. Springer, 2004, pp. 278–288.

[38] F.-L. Wong, F. Stajano, and J. Clulow, "Repairing the Bluetooth pairing protocol," in *Proceedings of International Workshop on Security Protocols*. Springer, 2005, pp. 31–45.

[39] Y. Shaked and A. Wool, "Cracking the Bluetooth PIN," in *Proceedings of the conference on Mobile systems, applications, and services (MobiSys)*. ACM, 2005, pp. 39–50.

[40] A. Y. Lindell, "Attacks on the pairing protocol of Bluetooth v2.1," *Black Hat USA, Las Vegas, Nevada*, 2008.

[41] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 447–462.

[42] Y. Burakova, B. Hass, L. Millar, and A. Weimerskirch, "Truck hacking: An experimental analysis of the SAE j1939 standard," in *10th USENIX Workshop on Offensive Technologies WOOT 16*, 2016.

[43] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile driver fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 1, pp. 34–50, 2016.

[44] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1044–1055.

[45] S. Nürnberger and C. Rossow, "Vatican vetted, authenticated CAN bus," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 106–124.

[46] J. Van Bulck, J. T. Mühlberg, and F. Piessens, "VulCAN: Efficient component authentication and software isolation for automotive control networks," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, 2017, pp. 225–237.

[47] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th USENIX Security Symposium 16*, 2016, pp. 911–927.

[48] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 1110–1115.

[49] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *2016 international conference on information networking (ICOIN)*. IEEE, 2016, pp. 63–68.

[50] K.-T. Cho and K. G. Shin, "Viden: attacker identification on in-vehicle networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1109–1123.

[51] Q. Hu and F. Luo, "Review of secure communication approaches for in-vehicle network," *International Journal of Automotive Technology*, vol. 19, no. 5, pp. 879–894, 2018.

[52] S. Kulandaivel, T. Goyal, A. K. Agrawal, and V. Sekar, "Canvas: Fast and inexpensive automotive network mapping," in *28th USENIX Security Symposium*, 2019, pp. 389–405.

[53] M. Marchetti and D. Stabili, "Read: Reverse engineering of automotive data frames," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1083–1097, 2018.

[54] M. D. Pesé, T. Stacer, C. A. Campos, E. Newberry, D. Chen, and K. G. Shin, "LibreCAN: Automated CAN Message Translator," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2283–2300.

[55] D. Frassinelli, S. Park, and S. Nürnberger, "I know where you parked last summer: Automated reverse engineering and privacy analysis of modern cars," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1401–1415.

[56] Google, "Android Auto," https://www.android.com/auto/, 2020.

[57] Apple, "CarPlay," https://www.apple.com/ios/carplay/, 2020.

[58] S. Mazloom, M. Rezaeirad, A. Hunter, and D. McCoy, "A security analysis of an in-vehicle infotainment and app platform," in *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, 2016.