A Non-Invasive Cyberrisk in Cooperative Driving

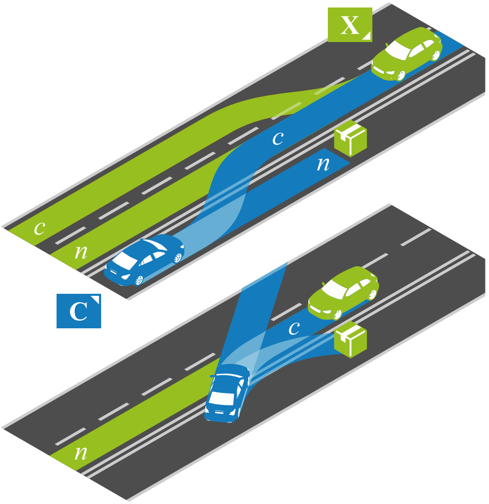
F. Bapp[1A], J. Becker[1A], J. Beyerer[1B,2], J. Doll[3], M. Filsinger[2], Ch. Frese[2],
Ch. Hubschneider[3], A. Lauber[1A], J. Müller-Quade[1C], M. Pauli[1D], M. Roschani[2],
O. Salscheider[1E,3], B. Rosenhahn[4], M. Ruf[2], Ch. Stiller[1E], D. Willersinn[2], J.R. Ziehn[2,4,†]

***Abstract*—This paper presents a hacking risk arising in fully-automated cooperative driving. As opposed to common cyberrisk scenarios, this scenario does not require internal access to an automated car at all, and is therefore largely independent of current on-board malware protection. A hacker uses a wireless mobile device, for example a hacked smartphone, to send vehicle-to-vehicle (V2V) signals from a human-driven car, masquerading it as a fully-automated, cooperating vehicle. It deliberately engages only in high-risk cooperative maneuvers with other cars, in which the unwitting human driver is expected to perform a specific maneuver to avoid collisions with other vehicles. As the human driver is unaware of the planned maneuver, he fails to react as expected by the other vehicles; depending on the situation, a severe collision risk can ensue. We propose a vision-based countermeasure that only requires state-of-the-art equipment for fully-automated vehicles, and assures that such an attack without internal access to an automated car is impossible.**

I. Introduction and State of the Art

While the field of "autonomous" fully-automated driving considers the automated vehicle in an uncontrollable, almost hostile environment, *cooperative* fully-automated driving introduces a fundamentally different paradigm. This paradigm generally extends the planning space from a single vehicle to several vehicles around it (cf. [FBWB08], [FB11b]), but at the same time provides handles to reduce the situation complexity and increase safety, by allowing individual vehicles to lay trust in the actions of other vehicles around them (e.g. [FB10], [FB11a], [TSP+17]).

Not in all, but in many key applications, this trust enables a reduction of safety margins. It can thereby provide safe solutions to otherwise highly risky situations (such as cooperative collision avoidance, e.g. [BWB09]) or significantly speed up everyday situations (such as merge or intersection scenarios).



(a) Initial situation upon obstacle detection, with best maneuver options in case of cooperation (*c*) and no cooperation (*n*).

(b) Situation when lack of cooperation is realized, with the remaining maneuver options, all of which involve significant collision risks.

Fig. 1: Example attack scenario. (a) The fully-automated vehicle (**C**, lower left) detects an obstacle ahead, while another vehicle (**X**, upper right) is approaching on the opposite lane. If **X** is non-cooperative (e.g. human-driven, case *n*), the safest solution for **C** is to brake within the current lane, possibly colliding with the obstacle at low speed. If **X** is instead cooperative (case *c*), then **C** and **X** can negotiate to both change lanes, to avoid the obstacle and each other. (b) In the case of the described hacking attack, **C** will assume cooperation from **X** (and therefore execute its *c* maneuver), while **X** is actually human-driven and non-cooperative, executing its *n* maneuver.

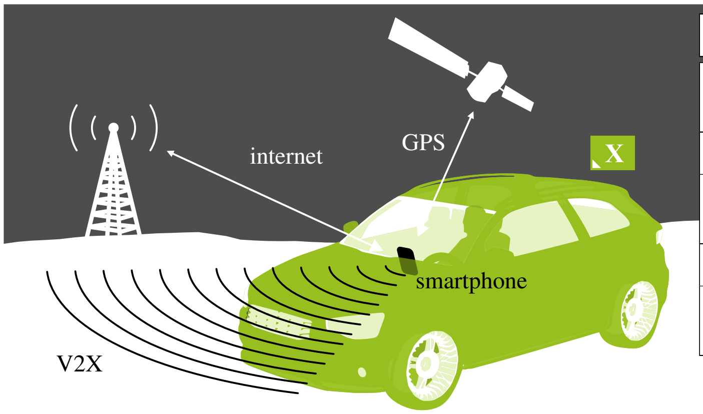
At the same time, trust always attracts attempts at exploiting it. In the case of safety-critical driving situations, a systematic violation of trust can cause severe accidents, obstruct traffic and damage the industry; inducing such effects can be the intent of various parties, in particular in the areas of national sabotage and terrorism.

Preventing such activities has been a subject of intensive research; overviews of cyberrisks for automated driving and corresponding countermeasures can be found in [YGA15], [Wei15], and in particular [PS15]. A common assumption is that cyberattacks on cooperative driving will require the attacker to manipulate an automated vehicle (for example by attacking the CAN bus, sensors, planning units or V2V communication units) or infastructure (for example by shutting down roadside infrastructure, manipulating signs, or manipulating online map databases). All of these attacks can have a severe impact on automated and cooperative driving; however, these attacks always involve manipulation (physical or remote) of a dedicated traffic system, such as an automated vehicle itself, roadside infrastructure, or online services. These traffic systems are clear targets for cyberattacks, but at the same time can (due to their defined application and limited number of providers) be effectively regulated, and are usually designed and maintained under high security standards.

In contrast, this paper will present an attack that does *not* require access to any dedicated traffic system, and for

*This publication was written in the framework of the Profilregion Mobilitätssysteme Karlsruhe, which is funded by the Ministry of Science, Research and the Arts and the Ministry of Economic Affairs, Labour and Housing in Baden-Württemberg and as a national High Performance Center by the Fraunhofer-Gesellschaft.

[1] Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany
[A] Institute for Information Processing Technologies (ITIV)
[B] Vision and Fusion Laboratory (IES)
[C] Competence Center for Applied Security Technology (KASTEL)
[D] Institute of Radio Frequency Engineering and Electronics (IHE)
[E] Department of Measurement and Control (MRT)
[2] Fraunhofer IOSB, 76131 Karlsruhe, Germany
[3] Research Center for Information Technology (FZI)
[4] Leibniz Universität Hannover, 30167 Hanover, Germany, Institut für Informationsverarbeitung (TNT)
[†] Corresponding author, jens.ziehn@iosb.fraunhofer.de



Source	Information
V2X	• environment state (other vehicles, sensor data, ...) • planned maneuver (**C**'s intentions)
GPS	• **X**'s current state (and **X**'s driver profile)
Internet	• environment state (traffic, road condition) • visibility (weather, position of the sun)
Inertial	• **X**'s current state (and **X**'s driver profile)
Device Services	• **X**'s intentions (if navigation is enabled) • distractions (active phone calls, music, ...)

TABLE I: Examples of data that can be used by the malware to actively engage in credible V2V communication and effect traffic accidents.[1]

this reason is not addressed by common countermeasures. In the most problematic variant, an attacker can cause a series of severe traffic accidents by infecting a significant portion of smartphones with malware, using the wireless network antennae to broadcast false vehicle-to-vehicle (V2V) messages in VANETs (vehicular ad-hoc networks). This attack pattern thus focuses on the near future introduction of fully-automated cooperative driving in mixed traffic (i.e. traffic with automated *and* non-automated vehicles), and is expected to gain relevance due to an extension of mobile phone frequencies towards V2V frequencies (cf. [LKE13], [PK14]) and the market penetration of software-defined radio in smartphones (cf. [Ram07]), making it increasingly likely that software attacks can allow smartphones to actively participate in V2V communication.

We will describe variants of the attack and the corresponding prerequisites in detail in Sec. II; Sec. III will demonstrate the threat on simulated scenarios; Sec. IV will propose a countermeasure based on providing a second and more robust authentication factor for a potentially cooperative vehicle and its intentions (i.e. in addition to V2V communication) through visible light communication (VLC), which only requires common automated vehicle equipment; Sec. V will summarize the main points and provide an outlook to future work.

II. Attack Overview and Risks

The attack, exemplified in Fig. 1, assumes mixed traffic of cooperative fully-automated and human-driven vehicles, and is more efficient when targeting a large number of "*direct victims*" at the same time. However, for simplicity, we will describe the attack on a single direct victim. The direct victim is the human driver of a non-automated and non-communicating vehicle, which is, through the attack, masqueraded as an automated cooperative or at least communicating vehicle. The *attacker* makes a device in the victim's car (possibly the victim's smartphone, infected with a malware) send V2X messages. These messages falsely identify the victim's car as fully-automated and cooperative.

When, for example in critical situations, other cooperative vehicles request a maneuver that involves the victim's car at narrow safety margins, the malware communicates agreement. Since the victim has no knowledge of the agreed maneuver, the victim likely does not act according to plan, and depending on the situation, an accident (involving the victim's car and other traffic participants, the *indirect* victims) can become very likely, as will be shown in Sec. III. In the following, the elements of such a scenario will be discussed in more detail.

A. The Direct Victim

The direct victim is the human driver of a non-automated car (**X** in Fig. 1), which is in particular incapable of V2X communication. He does, however, own a smartphone which is physically able to establish V2X connections (how and why this is possible will be outlined in Sec. II-C1).

B. The Attacker

The attacker intends to falsely identify the cars of one or more direct victims as cooperative fully-automated vehicles, to cause accidents involving these cars and possibly indirect victims (such as **C** in Fig. 1). For this, he must be technically able to construct a specifically-built device, or a smartphone malware, as outlined in Sec. II-C, and locate it in the car of one or several victims. Since a successful attack depends on both the malware and the traffic situation, it is difficult to target specific victims. For reasons that will be described in Sec. II-C2, the attacker only has a short time frame between the first attempted attack and the blocking of necessary certificates, which should effectively end the attack. The most likely case is an attacker trying to place malware copies in as many vehicles as possible, and then trigger all of them at the same time to cause random, untargeted accidents in a fraction of these victims. For these reasons, likely motives for the attack are cyberterrorism or national sabotage rather than common criminal activities.

C. The Malware/Device

For an effective attack, the malware/device must be able to participate technically in V2X communication (Sec. II-C1), authenticate itself as a valid vehicle (Sec. II-C2), and gain awareness of the car's situation, to assess (and potentially

[1] It should be noted that not all kinds of data will necessarily be accessible, depending on the device's specifications and the capabilities and access rights and design of the malware.

affect) its criticality, and remain undetected until a sufficiently critical opportunity is detected (Sec II-C3).

While a custom-made device planted in a vehicle (or in vehicle parts before assembly) can easily be designed to support V2V communication standards, foreseeable developments in wireless technology (namely extensions to the 802.11ac wireless standard and the introduction of software-defined radio) will likely enable smartphones to participate in V2V communication with mere software modifications by a suitable malware, leading to a considerably more immediate threat. Since the attack is most effective when a large number of drivers is affected at the same time (because only a fraction of the malware devices will be able to cause a relevant accident before its certificate is revoked, cf. Sec. II-C2), infecting smartphones is an appealing attack vector that allows to affect a large number of victims without requiring any physical presence. For this reason, we will focus in particular on the vulnerability of smartphones.

1) V2V Communication: The key prerequisite for the device is to be equipped with an antenna capable of engaging in V2X communication. While V2X communication is still under development, and hence standards may evolve before the general introduction of fully-automated cooperative driving, IEEE WAVE and DSRC are current standards supported by the U.S. Federal Communication Commission (FCC) and the European Telecommunications Standards Institute (ETSI) and will be discussed here in detail.

a) WAVE and DSRC: The current VANET standards are IEEE 802.11p (or “WAVE”, wireless access in vehicular environments), and its variation DSRC (dedicated short-range communications) or ETSI ITS-G5 (the corresponding standard in Europe). DSRC, as introduced by the FCC in 1999 (cf. [JTM^{+}06]), allocates 70 MHz to seven V2X communication channels as shown in Fig. 2. Channels 172 and 178 are restricted to safety communications, channels 174, 176, 180 and 182 are reserved for general communication, and channel 184 is reserved for high power, long range (“HP/LR”) communication with a maximum output power of 40 dBm.[2] To masquerade as a fully-automated vehicle, the device must be able to communicate on these channels. Most modern smartphones feature antennae for wireless communication using the standards 802.11a/b/g/n/ac; these standards are distinct from 802.11p and DSRC, but their frequencies are similar and, according to FCC plans for 802.11ac, may overlap in the future (cf. Fig. 2, [LKE13]).

b) Relation to Mobile Phone 802.11ac: The described overlap has negative impact on communication stability due to interference and competition between devices, as detailed in [PK14]; the relevant implication in terms of cyberattacks is, however, that mobile phones will have to support V2V frequencies to fully use 802.11ac. While the DSRC protocol differs from WLAN in various ways, including a narrower bandwidth per channel (10 MHz instead of 20 MHz to 160 MHz), whether these parameters are hard-wired or programmable via malware depends on the device specifications.

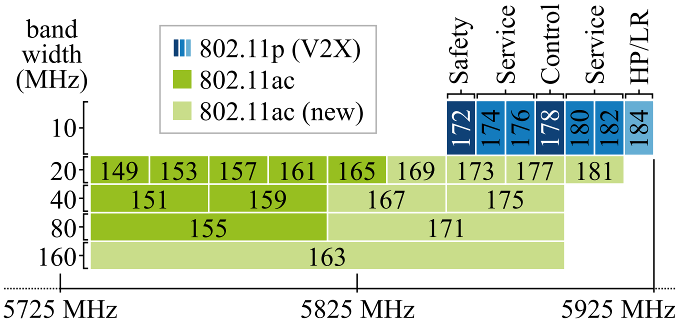


Fig. 2: Overview of relevant wireless channels according to IEEE 802.11: Current V2X channels in the 802.11p standard (DSRC/WAVE), and current WiFi channels in the 802.11ac standard, along with proposed new channels for 802.11ac according to FCC NPRM 13-22/49 (see [LKE13], [ENW^{+}14], [PK14]) overlapping with DSRC/WAVE. The channels are labeled with the center channel number, as well as FCC planned channel use for DSRC.

c) Software-Defined Radio: Software-defined radio (SDR) is an emergent technology that equips devices with programmable antennae, allowing them to use an even wider range of frequencies. Due to the increased flexibility with respect to varying communication standards, SDR is ascribed good prospects for the mobile phone market (see [Ram07]).

2) Authentication: In addition to being able to joining VANETs, the malware/device must be able to identify itself as a valid sender. VANET authentication systems are Public Key Infrastructures (PKI), like the U.S. Security Credential Management System (SCMS, cf. [WWKH13]), and the European V2X PKI (cf. [BSS^{+}12]). These infrastructures issue certificates to individual V2X radio devices, and messages received in VANETs are only trusted by the recipient if the sender has used a valid private key to sign the message, and provides a valid public key and a certificate. If the security of a unit is known to be compromised (as by the described hacking attack), the certificate is revoked by the PKI, and all V2X units are instructed not to trust this certificate anymore.

This poses two main challenges for the attacker: A valid certificate has to be obtained, and then maintained sufficiently long, by avoiding revocation until after a successful attack (which will be discussed in detail in the following Sec. II-C3). To establish a robust PKI, obtaining certificates and private keys has to be extremely difficult. One way to obtain certificates can be theft from a legitimate owner or authority, as was the case in the Stuxnet worm identified in 2010, which attacked Iran’s nuclear program by use of stolen certificates from two companies, JMicron and Realtek (see [MRHM10]); another way is *side-channel attacks*, in which an attacker monitors external properties in security hardware components (such as power consumption, electromagnetic leaks or sound) and possibly implementation details, and uses signal analysis to deduce information about hidden data, such as private keys (see [SLP06]). Side-channel attacks are widely neglected in the context of vehicles, with the common argument, that an attacker exploiting a side channel could as well manipulate the

[2] As opposed to 28.8 dBm for channels 172 through 178, and 20 dBm for the remaining channels, see [FCC04].

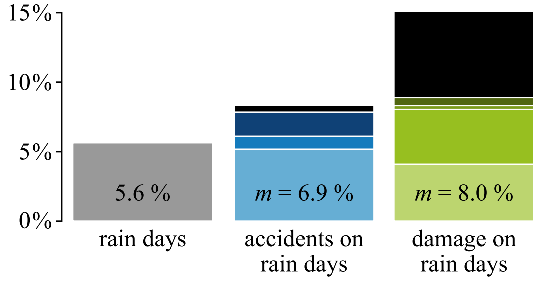


Fig. 3: Severity of accidents by weather, using the example of a total of 2 389 accidents recorded by the Karlsruhe police on seven road segments in and around Karlsruhe (B10, B36, Kapellenstr. and Adenauerring, grouped and colored by segment) between 2012 and 2016, matched to data of the Deutscher Wetterdienst (German Meteorological Office). While only 5.6 % of days had more than 1 mm of precipitation, both the mean accident count (+22 %), and the accident damage (+41 %), were significantly higher on these days.

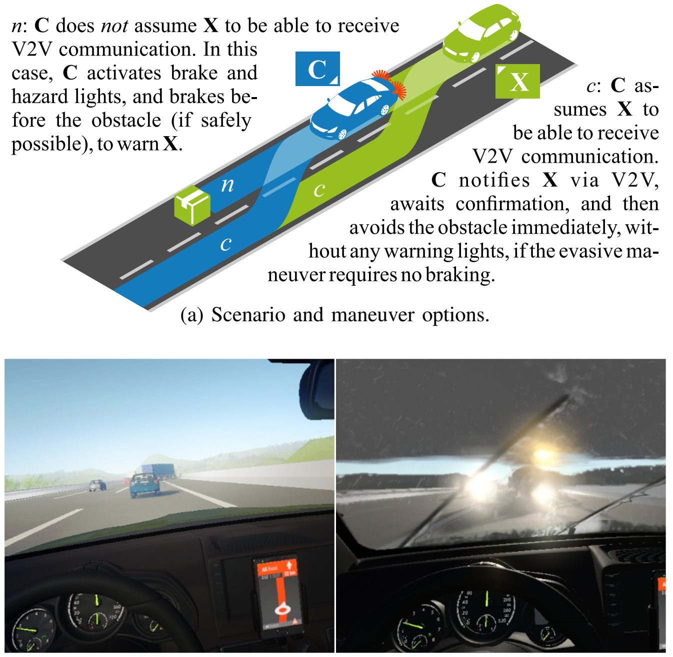
brakes. This paper shows that this argument is flawed, because the attack affects a different car.

In either case, certificates are relatively difficult to obtain, and will be available in limited numbers. Since any suspicious activity, and in particular any successful attack, significantly increases the risk of having one or all stolen certificates revoked, the attacker would have a very limited timeframe to execute the attack, namely between the first recognized attempt, and the notification of all (or most) cooperative vehicles of the revoked certificates. When using smartphones, it is therefore likely that an attack would not be triggered upon installation of the malware on one phone, but by a remote broadcast to all malware devices when a significant spread of the malware is assumed by the attacker, such that the entire attack takes place within hours, not days.

3) Situation Awareness: To believably participate in V2V communication (and thus avoid certificate revocation before an opportunity for a serious attack) and to estimate the criticality of maneuvers, the device can benefit considerably from inertial sensors, GPS and an internet connection, which are all present in modern smartphones. Table I gives an overview of the main information sources that can be made available to a device, and in particular to smartphone malware, allowing it, for example, to approximately reconstruct the trajectory of **X**, learn driver behavior, collect weather or traffic information from the internet and use distractions, for example passively when the driver is having a phone conversation, or actively by ringing alarms. Section II-D and Figs. 3 and 4 indicate how such information can be used to more effectively judge criticality and increase the likeliness of accidents in an attack.

D. Types of Attacks

The device can be programmed to perform various kinds of attacks. These fall into two main groups: Attacks involving fully-automated vehicles capable of cooperative maneuver planning, and attacks involving (not necessarily “fully”-) automated vehicles capable of V2V *communication*, but *not cooperation*. Examples of the former have already been stated, namely agreeing to potentially risky maneuvers with narrow safety margins, in which an unwitting human driver **X** would likely cause an accident.



n: **C** does *not* assume **X** to be able to receive V2V communication. In this case, **C** activates brake and hazard lights, and brakes before the obstacle (if safely possible), to warn **X**.

c: **C** assumes **X** to be able to receive V2V communication. **C** notifies **X** via V2V, awaits confirmation, and then avoids the obstacle immediately, without any warning lights, if the evasive maneuver requires no braking.

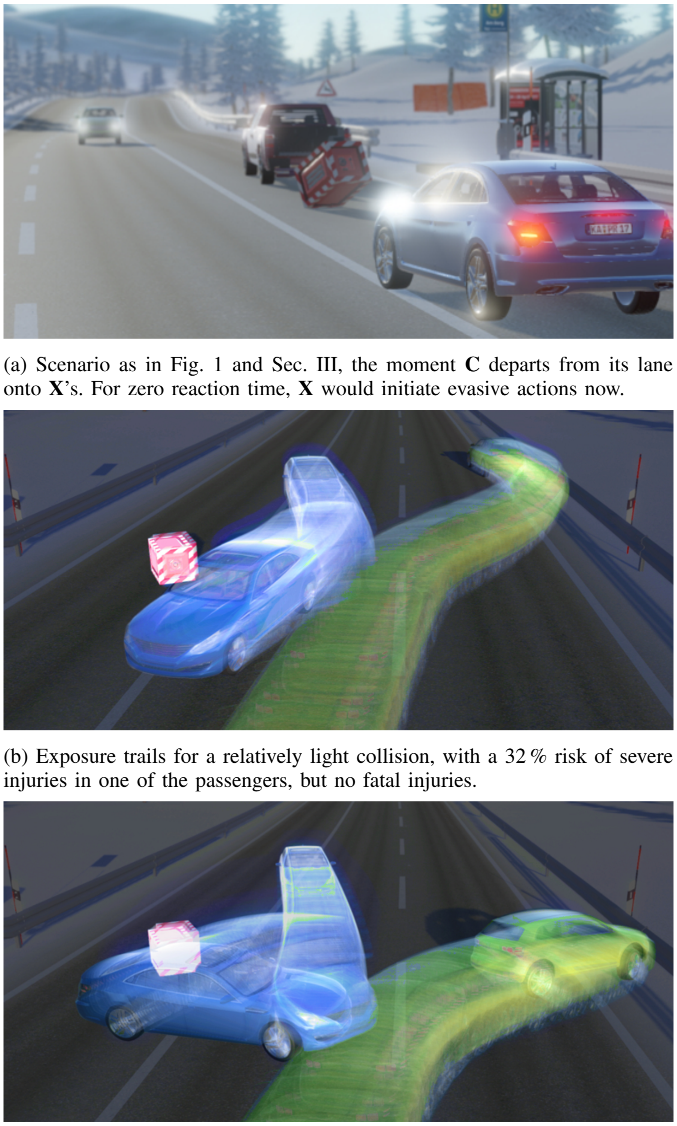
(a) Scenario and maneuver options.

(b) Scenario as seen by **X**’s driver in a virtual reality simulation. Left: During good weather, the obstacle (a broken down truck) is easily seen, even without explicit warning from **C** (ahead). Right: With rain and blinding from a low sun, the situation is considerably more challenging for **X**.

Fig. 4: Examplary simulation results for the scenario described in Sec. II-D, where the malware suppresses protective behavior of the automated vehicle towards the human driver by pretending it receives V2V communication.

Attacks which masquerade a vehicle as non-cooperative but communicating again fall into two main categories. In the first, the malware *communicates a specific but false data*, such as alarming rear automated vehicles of an imminent emergency brake or obstacles on the road (causing them to execute risky evasive maneuvers).[3] In the second, the malware *suppresses behavior* that other vehicles would normally exhibit towards human drivers. For example, it is expected that if a fully-automated vehicle avoids an obstacle, and is followed by non-cooperating (possibly human driven) vehicles, the fully automated vehicle will not only avoid the obstacle, but also take measures to warn the human driver, such as flashing the hazard lights, enabling the brake lights immediately upon detection of the obstacle (even if the automated vehicle does not brake), or braking slowly, possibly even to a halt, instead of evading immediately. A cyberattack however (as shown in Fig. 4), can cause a fully-automated cooperative vehicle to avoid an obstacle *without* warning the human driver in rear, because the malware informed the rear vehicle via V2V about the obstacle, and received confirmation. As **X** thus receives no warning at all, the collision risk is considerably increased, in particular in challenging situations: As shown in Tab. I, the malware can use detailed weather information online to estimate the criticality of the situation. Even a rough

[3] It should be noted that causing accidents by sending false sensor data is in general possible, but difficult for a malware that has no direct environment perception, and thus cannot determine what dangerously false sensor information would be.



(a) Scenario as in Fig. 1 and Sec. III, the moment **C** departs from its lane onto **X**'s. For zero reaction time, **X** would initiate evasive actions now.

(b) Exposure trails for a relatively light collision, with a 32 % risk of severe injuries in one of the passengers, but no fatal injuries.

(c) Severe collision, with a 86 % risk of severe injuries in at least one of the passengers, 44 % risk of fatal injuries for passengers in **X**, and 14 % for **C**.

Fig. 5: Examplary simulation results for the scenario of Sec. III simulated in OCTANE[4]. Injury estimates should be regarded as rough approximations.

comparison based only on daily weather records given in Fig. 3 shows a significant increase in accidents and accident severity on days with notable precipitation. These results are matched by simulation experiments with humans driving in virtual reality (Fig. 4), who were able to resolve the above situation successfully during clear daytime conditions even with a malware attack, but caused accidents or near-accidents under challenging weather conditions.

III. Simulation Results

The described attack pattern has been evaluated in detail in a simulation scenario based on the introductory example of Fig. 1, where the automated vehicle avoids an obstacle by changing onto the oncoming lane, expecting the oncoming vehicle to clear the lane, as negotiated via V2V. The simulation is set up with three vehicles:

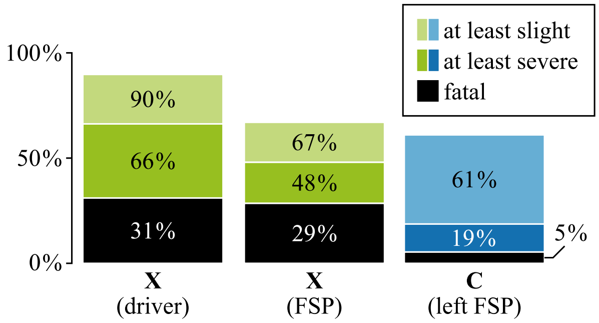


Fig. 6: Estimated severities over simulated crashes as described in Sec. III according to [RC09], with probabilities of injury types for the passengers of both cars. Since the crash simulation has very limited accuracy, these estimates are only rough approximations. For **X**, the driver and the front seat passenger (FSP) are given separately, as there is a notable difference between them. The driver of **X** is affected more by side collisions with **C**, while the FSP is affected mostly by collisions with the guardrail (cf. Fig. 5). **C** is generally affected less, partly due to the lower initial speed, but also due to the better reaction that reduces speeds faster and avoids side impacts.

A pickup, which loses its load (the obstacle) in front of a fully-automated cooperative vehicle **C**. The behavior of the pickup is constant throughout this evaluation, and unrelated to the scenario except for providing a control reference for **C**'s ACC, and for losing its load at a predetermined time.

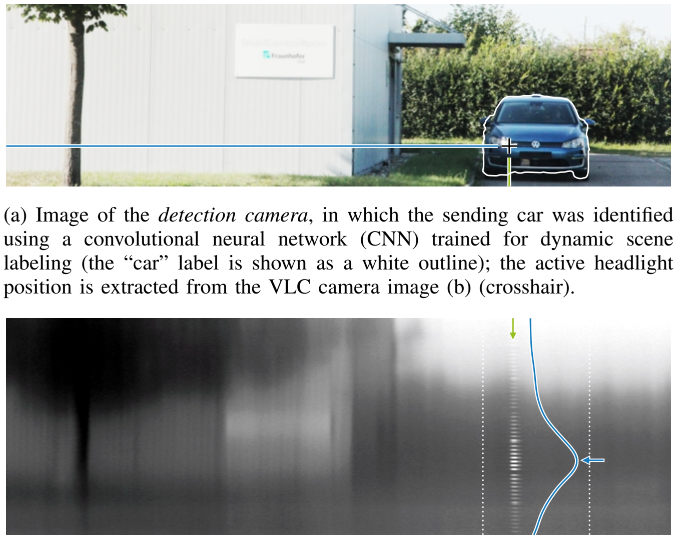
A fully-automated cooperative vehicle C, which has to evade the obstacle. The behavior of **C** is determined by a maneuver planner which provides ACC (initial state), evasion by lane change, or emergency braking within the lane (depending on the V2V communication channel). It uses a combination of LIDAR, camera and stereo obstacle detection.

The victim's vehicle X, whose malware device offers to change lanes without informing the human driver. As **X** is, in the attack scenario, driven by an unwitting human driver, the behavior of **X** is given by a sudden evasive reaction when the driver recognizes that **C** has entered **X**'s lane (after a variable reaction time t_{react}). For comparison, the ideal case of **X** being fully-automated and cooperating is simulated as well.

The scenario is evaluated parametrically, in the sense that several parameters that determine the outcome are tested. Therefore, the behavior of **X** in the simulation is specified by a parametric model, not by a human in the loop. Specifically, the evaluated parameters are **X**'s reaction time (between **C** departing from its lane onto **X**'s, Fig. 5a, and **X**'s first evasive reaction), **X**'s reaction type (steering function, braking function), **C**'s maneuver, and road friction.

Brake reaction times were varied based on [JR71], which places brake reaction times roughly between 0.5 seconds and 1.1 seconds (real scenarios with drivers expecting the brake event over a length of 10 km), and [MMB00], which gives reaction times between 0.96 and 1.28 seconds in an unexpected incursion scenario, both simulated and real. The reaction time of **X** had the most significant impact, with simulations showing that above a reaction time of about 0.7 seconds, a collision between **X** and **C** is inevitable. The severity is

[4] www.octane.org



(a) Image of the *detection camera*, in which the sending car was identified using a convolutional neural network (CNN) trained for dynamic scene labeling (the “car” label is shown as a white outline); the active headlight position is extracted from the VLC camera image (b) (crosshair).

(b) Image of the *VLC camera*, showing the car boundaries projected from the detection camera image (a), and the signal pattern column produced by the headlight. The horizontal position of the column, and the vertical intensity profile (blue) are used to estimate the position of the sending headlight, to match it with the camera image in (a). Figure 8 shows the signal processing.

Fig. 7: Example transmission of a looped, Manchester-encoded, binary $\mathbf{1}^8\mathbf{0}^8$ pattern. The sender light is mounted to a stationary VW e-Golf 7, the detection camera (image in (a)) and the receiver camera (image in (b)) are positioned behind the windscreen of a VW Passat B7 approaching at 20 km/h.

difficult to predict from the parameters, because the distinction between a head-on collision and a side collision, as well as the impact strength, depend on small variations in maneuver execution. Since head-on collisions are generally safer, faster steering reaction times do not always result in a lower accident severity. Figure 6 gives a distribution of the estimated accident outcomes over all reaction times greater than 0.7 seconds; since the simulation does not use detailed collision models, these estimates should be treated with caution; however, they can hint at the actual severity.

IV. Countermeasure

The particular threat of the described attack scenario lies in the fact that the device carrying the malware is not a high security system, but either a separate, specifically-built device, or a generic mobile phone, which are considerably more vulnerable and difficult to protect as a whole.

For this reason, the proposed countermeasure does not aim at preventing smartphones from being hacked, or from establishing V2V connections, but instead at assuring that a maneuver negotiated via V2V was correctly received by the control module of each vehicle. This is achieved by implementing an additional protocol, by which any negotiated maneuver must be confirmed via *visible light communication* (VLC), namely sending information through pulses of the head or tail lights, that are invisible to the human eye, but can be recorded by on-board cameras of other cooperative vehicles, and associated visually with the sender, to validate the source. Thereby, a similar hacking attack would require seizing control of the victim car’s light control system, and thereby fall into the same prevention category as classic sabotage attempts, mitigating the risk of a non-invasive attack as outlined here.

A. Visible Light Communication

Automotive LED lights can be pulsed with high frequencies, to adjust the intensity, which is perceived as constant when the pulse frequency exceeds the *flicker fusion rate* of the human eye (which varies depending on the particular conditions, and in particular the ratio between high and low intensities). Since visible flicker can be unpleasant, distracting or harmful (cf. [IEE15]), light pulse communication must aim to be invisible to the human eye. Pulsing LEDs with several 1000 Hz is not problematic—the bottleneck instead is the recording camera’s frame rate. For usual automotive applications, camera frame rates rarely exceed 100 Hz; cameras with high frame rates quickly become prohibitively expensive, and are not expected to be widely installed in production-line automated vehicles, for a lack of other relevant applications. Therefore, the proposed countermeasure instead exploits the *rolling shutter effect* that is found in the majority of CMOS camera sensors (with the exception of specialized *global shutter* sensors), but usually considered an undesirable yet tolerable side effect. With rolling shutter, sensor pixel intensities are only measured simultaneously over a single *line* of pixels, while different lines are read (and reset) in progressive order, leading to slightly offset exposure intervals for each line of pixels. The effect is usually negligible for slow-moving scenes, but can cause artifacts or deformations when the scene changes rapidly.

In the proposed countermeasure, the rolling shutter effect is used to capture light pulse signals at a rate far higher than the frame rate of the camera (in the implementation in Fig. 7, more than 80 times), while retaining spatial information necessary to match the signal pattern to the lights of a specific car, to assure the light pulse signal is sent by the correct car’s lights. To this end, a rolling shutter camera is equipped with an optical low-pass filter that convolves the image perpendicularly to the sensor line direction, thereby distributing intensity of pointlike car lights across all pixel lines. If the car lights send a pulse signal, the signal can be extracted from the rolling shutter

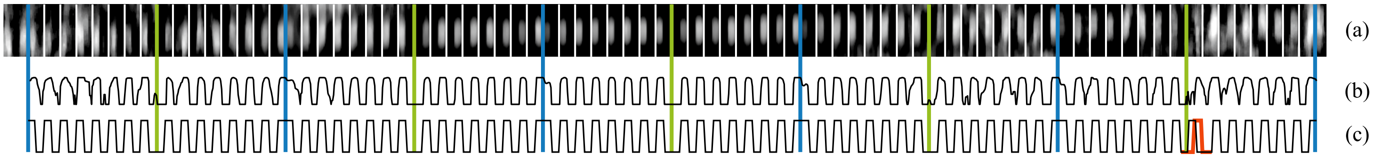


Fig. 8: Extracted and rotated signal line from Fig. 7, showing the post-processed intensities (a), the scoring function (b) and the extracted signal (c) containing one erroneous (flipped) bit. Manchester pairs, byte breaks and sequence breaks are indicated in (a), the erroneous bit is highlighted in (c).

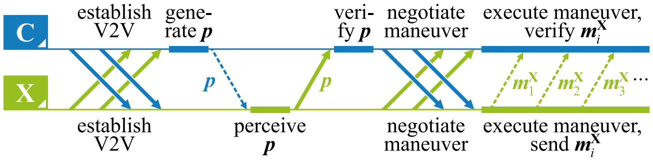

C
X
establish V2V
generate $\boldsymbol{p}$
p
verify $\boldsymbol{p}$
p
negotiate maneuver
execute maneuver, verify $\boldsymbol{m}_i^{\mathbf{X}}$
$m_1^{\mathbf{X}}$
$m_2^{\mathbf{X}}$
$m_3^{\mathbf{X}}$ …
establish V2V
perceive $\boldsymbol{p}$
negotiate maneuver
execute maneuver, send $\boldsymbol{m}_i^{\mathbf{X}}$

Fig. 8: Proposed VLC verification protocol between two cars, **C** and **X** (shown only from the perspective of **C** verifying **X**; the actual process is symmetric), with solid arrows (⟶) for VANET communication, and dashed arrows (⇢) for VLC. The first message, $\boldsymbol{p}$, validates the ability of **X** to perceive VLC. During cooperative maneuver planning, **X** is assigned a maneuver sequence $(\boldsymbol{m}_i^{\mathbf{X}})_{i\in\mathbb{N}}$ which encodes both the agreed maneuver for **X** and random information. During maneuver execution, **X** continuously broadcasts the sequence via VLC, which is verified by **C**, leading to an exponential gain in trust over time.

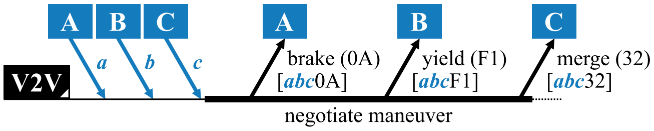


Fig. 9: Simplified example of a way to establish the code sequences $(\boldsymbol{m}_i^{\kappa})_{i\in\mathbb{N}}$ used in Fig. 8 via VANET (labeled "V2V"), for cooperative participants $\kappa \in \{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$. Each participant contributes a randomly generated symbol $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$. Then the maneuver is negotiated. Each participant in this process receives a maneuver role (here brake, yield, merge), each of which is associated with another symbol (here shown as one byte in hexadecimal representation behind the role). By combining the random symbols and the individual role symbols (shown in the brackets), and generating a hash value over the combined symbols, all participants can compute a unique number for each participant. These numbers can be used as seeds for a common pseudorandom generator function, that provides each participant with a message sequence to send, and the corresponding message sequences to expect from each other.

timing. At the same time, the car light location can be extracted with good accuracy, and can be matched with other sensor data to validate the sender's position, as shown in Figs. 7 and 8. All information sent via VLC is encoded using *Manchester code* (see [CCC+14]), in which a binary `1` is encoded as `10`, and a binary `0` as `01`, resulting in a constant mean intensity.

B. Outline of the Proposed Protocol

The proposed protocol intends to establish reasonable trust before a possible "point of no return", but not necessarily before the maneuver is executed. This is based on the rationale that maneuver criticality is usually not constant, but increases over time. For example in the situation in Fig. 1, the maneuver part performed by **C** before entering the opposite lane (in the simulation of Sec. III around 0.54 s) is largely independent of whether **X** is cooperative or human-driven. The maneuver can be aborted safely for about one second before a point of no return is reached.[5]

The protocol, shown in Fig. 8, consists of two separate validations, a *perception* validation and a *maneuver* validation, both of which use the process of communicating a sequence via VLC and comparing it with VANET messages. As with Fig. 8, we describe the process from the perspective of **C** verifying whether **X** is truly cooperative, keeping in mind that the same process will be initiated by **X** in reverse.

Perception. After a VANET connection is established between **C** and **X**, **C** generates a random message $\boldsymbol{p}$ and broadcasts it via VLC. **X** must perceive and extract $\boldsymbol{p}$, and communicate it back to **C** via VANET. This verifies that **X** is capable of perceiving the VLC code, ruling out cases in which **X** can *send* VLC, but receive only VANET.[6]

Maneuver. After a common maneuver has been negotiated via VANET, each participant receives a code sequence, which we denote $(\boldsymbol{m}_i^{\mathbf{X}})_{i\in\mathbb{N}}$ in the case of **X**. The code sequences are known among all VANET listeners. They encode both random information available *only* to VANET listeners (contributed by all maneuver participants), and information describing the individual maneuver or role of every participant. An example of a process to establish such a sequence is given in Fig. 9, which should be regarded as a simple illustration of the principle rather than a proposal for an actual implementation. When the maneuver begins, all participants broadcast their sequence over time, and verify the sequences broadcasted by other participants. If any participant (here **X**) is seen to broadcast a false sequence, the observer can terminate the cooperative maneuver, replace **X**'s motion prediction with a generic non-cooperative model, and notify all participants via VANET. Thereby, trust is established incrementally, with each correctly perceived bit halving the probability of a vehicle sending correct codes based on a false maneuver understanding.

Assuming only 20 Bit/s of effective information, the scenario in Sec. III would have a reduced hacking risk of 1 in 1700 before **C** entered the opposite lane, and 1 in 10^6 before the point of no return. The scenario in Fig. 4 would have been avoided almost certainly, because the lack of communication in **X** could have been noticed before the actual emergency. Since any cooperative maneuver can be assigned an estimated safety criticality (e.g. the safety margins or relative speeds), the protocol allows to tolerate relatively low trust levels during non-critical maneuvers, while aborting other maneuvers at the same trust level when the maneuvers involve collision risks.

C. Trust Gain

The authentication via VLC improves the trust with respect to pure V2V communication despite its relatively small bandwidth, because it is in several ways complementary to V2V. While a risk in V2V communication (exploited not only in this attack) is its undirectedness, VLC signals are immediately connected to the sender. Hence, to fake a VLC message, it is no longer sufficient to sham the message content, but also the method of broadcast (and, due to the proposed protocol, reception). Equipping a regular vehicle with the ability to send

[5] Maneuvers in which the distinction between a true cooperative vehicle and non-cooperative vehicle is critical, but cannot be sufficiently determined, should always be executed assuming the more conservative model of non-cooperation. For the given protocol, this includes maneuvers that become critical before a line of sight between the participants has been established for a sufficiently long time. In general, any maneuver planning process (for the VLC or any other verification method) should always consider not only the physical capabilities of the vehicles involved, but also whether sufficient trust can be established in time during the maneuver.

[6] Depending on the VLC modulation process, it could for example be possible for the device to vary the car's power consumption and thereby let the headlights' intensity flicker as desired; the device could also be integrated in car lights. Since perception of VLC is usually technically more advanced than emission (in particular if all technology has to remain hidden), verifying both capabilities increases the technical challenge for an attacker significantly.

light pulses is difficult: Using the actual headlights would effectively require an invasive attack; using other lights would require placing a light emitter on the vehicle that remains undetected *and* is regarded as a potential headlight position by the vision system of the receiver (cf. Fig. 7).

V. Conclusion and Outlook

This paper has presented a cyberrisk for fully-automated cooperative driving, in which a hacker can cause severe accidents without internal access to any dedicated traffic system, neither automated vehicles, nor infrastructure or dedicated network services or databases. Thereby, the attack circumvents all security measures taken to protect these systems from direct manipulation. Based on the current plans of technical developments in wireless technology and vehicle-to-vehicle (V2V) communication, it is expected that in the next few years, such an attack will be technically possible by just a malware installed on common modern smartphones. The attack has been analyzed both for its technical feasibility, and, by simulations, its ability to cause severe traffic accidents.

To counter the threat, this paper proposed establishing a second authentication factor in V2V communication, using visible light communication (VLC) to communicate messages between vehicles through flicker patterns of head and tail lights. A prototype implementation of the protocol, which only requires standard equipment in automated vehicles, was provided and tested both in simulated and real scenarios. This solution requires an attacker to either find a way to add VLC technology to a car without leaving visible traces, or to manipulate the automated system internally—effectively making the attack as difficult as (but less effective than), for example, hacking the maneuver planning mechanism directly.

Outlook

While the potential risks of the cyberattack are considered to be relatively well-understood, the proposed countermeasure is still under development, both in terms of its communication protocol and its technical realization, with the goal to establish a robust and unified second factor for authenticating VANET communications with standard vehicle technology.

References

[BSS$^+$12] N. Bißmeyer, H. Stübing, E. Schoch, S. Götz, J.P. Stotz, and B. Lonc. A generic public key infrastructure for securing Car-to-X communication. In *World Congress on Intelligent Transport Systems (ITS)*, pages 12–24, October 2012.

[BWB09] T. Batz, K. Watson, and J. Beyerer. Recognition of dangerous situations within a cooperative group of vehicles. In *2009 IEEE Intelligent Vehicles Symposium*, pages 907–912, Jun 2009.

[CCC$^+$14] A.M. Cailean, B. Cagneau, L. Chassagne, M. Dimian, and V. Popa. Miller code usage in Visible Light Communications under the PHY I layer of the IEEE 802.15.7 standard. In *2014 10th International Conference on Communications (COMM)*, pages 1–4, May 2014.

[ENW$^+$14] Sh.S. Elias, M. Nazri, D.M. Warip, R.B. Ahmad, and A.H. Abdul Halim. A Comparative Study of IEEE 802.11 Standards for Non-Safety Applications on Vehicular Ad Hoc Networks: A Congestion Control Perspective. In *Proceedings of the World Congress on Engineering and Computer Science 2014 Vol II WCECS 2014*, San Francisco, USA, Oct 2014.

[FB10] Ch. Frese and J. Beyerer. Planning cooperative motions of cognitive automobiles using tree search algorithms. In R. Dillmann, J. Beyerer, U.D. Hanebeck, and T. Schultz, editors, *KI 2010: Advances in Artificial Intelligence: 33rd Annual German Conference on AI, Karlsruhe, Germany, September 21–24, 2010. Proceedings*, pages 91–98. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[FB11a] Ch. Frese and J. Beyerer. Collision Avoidance by Cooperative Driving Maneuvers. *ATZelektronik worldwide eMagazine*, 6(5):48–52, Oct 2011.

[FB11b] Ch. Frese and J. Beyerer. A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 1156–1162, Jun 2011.

[FBWB08] Ch. Frese, T. Batz, M. Wieser, and J. Beyerer. Life cycle management for cooperative groups of cognitive automobiles in a distributed environment. In *2008 IEEE Intelligent Vehicles Symposium*, pages 1125–1130, Jun 2008.

[FCC04] U.S. Federal Communication Commission FCC. *FCC Record: A Comprehensive Compilation of Decisions, Reports, Public Notices, and Other Documents of the Federal Communications Commission of the United States*. Number 3 in 19. Federal Communications Commission, 2004.

[IEE15] IEEE. IEEE Recommended Practices for Modulating Current in High-Brightness LEDs for Mitigating Health Risks to Viewers. *IEEE PAR 1789*, 2015.

[JR71] G. Johansson and K. Rumar. Drivers' brake reaction times. *Human factors*, 13(1):23–27, 1971.

[JTM$^+$06] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich. Design of 5.9 GHz DSRC-based vehicular safety communication. *IEEE Wireless Communications*, 13(5):36–43, Oct 2006.

[LKE13] J. Lansford, J. B. Kenney, and P. Ecclesine. Coexistence of unlicensed devices with DSRC systems in the 5.9 GHz ITS band. In *2013 IEEE Vehicular Networking Conference*, pages 9–16, Dec 2013.

[MMB00] D.V. McGehee, E.N. Mazzae, and G.H.S. Baldwin. Driver reaction time in crash avoidance research: Validation of a driving simulator study on a test track. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 44, pages 3–320. SAGE Publications Sage CA: Los Angeles, CA, 2000. 20.

[MRHM10] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho. Stuxnet under the microscope. *ESET LLC (September 2010)*, 2010.

[PK14] Y. Park and H. Kim. On the coexistence of IEEE 802.11ac and WAVE in the 5.9 GHz Band. *IEEE Communications Magazine*, 52(6):162–168, Jun 2014.

[PS15] J. Petit and S.E. Shladover. Potential Cyberattacks on Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):546–556, Apr 2015.

[Ram07] U. Ramacher. Software-Defined Radio Prospects for Multistandard Mobile Phones. *Computer*, 40(10):62–69, Oct 2007.

[RC09] D. Richards and R. Cuerden. The Relationship between Speed and Car Driver Injury Severity. Technical report, Transport Research Laboratory, U.K. Department for Transport, Apr 2009.

[SLP06] K. Schramm, K. Lemke, and Ch. Paar. Embedded Cryptography: Side Channel Attacks. In K. Lemke, Ch. Paar, and M. Wolf, editors, *Embedded Security in Cars*, pages 187–206. Springer, 2006.

[TSP$^+$17] Ö.S. Tas, N.O. Salscheider, F. Poggenhans, S. Wirges, C. Bandera, M.R. Zofka, T. Strauss, J.M. Zöllner, and Ch. Stiller. Making Bertha Cooperate – Team AnnieWAY's Entry to the 2016 Grand Cooperative Driving Challenge. *IEEE Transactions on Intelligent Transportation Systems*, Nov 2017.

[Wei15] A. Weimerskirch. An Overview of Automotive Cybersecurity: Challenges and Solution Approaches. In *Proceedings of the 5th International Workshop on Trustworthy Embedded Devices*, TrustED '15, pages 53–53, New York, NY, USA, 2015. ACM.

[WWKH13] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn. A security credential management system for V2V communications. In *2013 IEEE Vehicular Networking Conference*, pages 1–8, Dec 2013.

[YGA15] E. Yağdereli, C. Gemci, and A.Z. Aktaş. A study on cybersecurity of autonomous and unmanned vehicles. *The Journal of Defense Modeling and Simulation*, 12(4):369–381, 2015.