

Security of Emergent Automotive Systems: A Tutorial Introduction and Perspectives on Practice

Anthony Lopez, *Student Member, IEEE*, Arnav Vaibhav Malawade, *Student Member, IEEE*,
Mohammad Abdullah Al Faruque, *Senior Member, IEEE*, Srivalli Boddupalli, *Student Member, IEEE*,
and Sandip Ray, *Senior Member, IEEE*

Abstract—Automotive systems are dominated by electronic and software components. A modern car includes hundreds of Electronic Control Units connected to a variety of diverse sensors and actuators, controlled by several hundred megabytes of software code and coordinated through several in-vehicle networks implementing disparate protocols. Given this complexity, such systems are unsurprisingly riddled by errors and vulnerabilities that can be effectively exploited to compromise their functionality in-field. Furthermore, our move towards more and more autonomous vehicles is accompanied by a sharp increase in design complexity, vulnerabilities, and attack surfaces. While there has been significant activity in automotive security research in recent years, a big gap remains between the state of the research and the needs of the practice. Furthermore, the research often does not account for the complexities of flow, methodologies, and dependencies that must be handled by practically viable solutions. In this paper, we address this problem by providing a unified, global view of the state of the research and practice in automotive security. We consider several general taxonomies of security threats, approaches explored in research to address them, and the state of the practice. Salient requirements for viable security solutions in current and emergent systems are discussed, and limitations of both research and practice are analyzed.

Index Terms—autonomous vehicles; connected cars; cybersecurity; security validation

I. INTRODUCTION

The last few decades have seen a transformation in automotive systems from mechanical or electro-mechanical systems to electronic, software-based systems. Modern automotive systems are complex distributed systems involving the coordination of hundreds of Electronic Control Units (ECUs) communicating through a variety of in-vehicle networks and the execution of several hundred Megabytes of software. However, they induce two additional constraints that result in significant design complexities beyond traditional distributed systems. First, the systems are cyber-physical: the ECUs coordinate, monitor, and control a variety of sensors and actuators including LIDAR, cameras, radar, light matrices, devices for sensing angular momentum of the wheels, devices for automated brake and steering control, etc. Second, many computation and communication tasks across the different ECUs,

sensors, and actuators must be accomplished under hard real-time requirements; *e.g.*, a pedestrian detection algorithm must complete a slew of complex activities including the capture of sensory data, aggregation, communication, analytics, image processing, security analysis, etc., within the time constraints to enable successful completion of the appropriate actuarial response such as warning the driver or automatically braking. Furthermore, the complexity is anticipated to rise sharply with increasing autonomy levels in vehicles. For instance, a future self-driving car with autonomy level 4 will include several elements not available in today's (level 2) systems. Example elements include: 1) vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications with a variety of networks of different levels of trustworthiness, 2) a diversity of sensors to detect driving conditions (*e.g.*, potholes, moisture, pedestrians, etc.), and 3) distributed computing elements to perform in-vehicle analytics and react to evolving conditions on the fly.

Security is obviously of paramount importance for automotive systems. Given that the system involves the complex interaction of sensory, actuarial, and computational elements, an “innocent” misconfiguration or error in one component may result in a subtle vulnerability that can be exploited in-field with potentially catastrophic consequences. Recent work has shown that it is viable, — and even relatively straightforward, — to hack a vehicle remotely, get control over its driving functionality, and cause an accident. The situation will be exacerbated in the future with the increase in autonomy level and the reliance on sensors and communications: an attacker may hack a vehicle remotely through the interception or tampering of sensor data and/or V2V and V2I messages without requiring physical access or even proximity to the vehicle under attack, thus resulting in a sharp increase in the attack surface. Consequently, the proliferation and adoption of autonomous, self-driving cars critically depends on our ability to ensure that they perform securely in a potentially adversarial environment. Unsurprisingly, there has been a large interest in recent years in the security of automotive systems, with a flurry of publications demonstrating a diversity of security vulnerabilities and exploits, as well as techniques for defense against these vulnerabilities.

Unfortunately, in spite of this interest, there has been little effort to consolidate, structure, and unify this large body of research. Consequently, publications in the area typically appear isolated approaches for specific attacks or defenses, rather than a disciplined study of security challenges or

A. Lopez, A. V. Malawade, and M. Al Faruque are with the Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA 92697, USA. Email: anth10@uci.edu, malawada@uci.edu, alfaruqu@uci.edu.

S. Boddupalli and S. Ray are with the Department of Electrical and Computer Engineering, University of Florida at Gainesville, Gainesville, FL 32611, USA. Email: bodsrivalli12@ufl.edu, sandip@ece.ufl.edu.

systematic approaches to counter them. Furthermore, much of the research on automotive security is conflated with other related areas on security assurance with analogous but different challenges, including wearables, the Internet-of-Things, or even traditional hardware and software designs. Finally, there has been an increasing divergence between academic research and industrial practice in the area, each of which has evolved independently with little interaction and in some cases with little understanding of the assumptions, issues, trade-offs, and scales considered by the other. All this leaves a researcher getting initiated in this area with the daunting tasks of sifting through the various challenges, complexities, and research directions; identifying approaches applicable to automotive systems in particular; and comprehending evolving challenges caused by the rising complexity of these systems through the past, present, and future.

This paper represents a first step to address the above problem. Our goal is to provide a comprehensive, systematic overview of both research and practice in automotive security. We develop a systematic categorization of research advances in various aspects of both attacks and defenses on automotive electronics. Furthermore, we discuss current practice in security assurance, point out the constraints and trade-offs, and provide perspective on the rationale involved. Our objective is for this paper to be a comprehensive article for a researcher to begin investigation on all aspects of the security of connected and autonomous vehicles.

The remainder of the paper is organized as follows. Section II provides a brief overview of automotive electronics and software to provide a sense of the complexity, potential vulnerabilities, and attacker entry points. We also recount different components of electronic and software system security, to provide a background for the rest of the paper. In Section III, we discuss some challenges in mitigating security vulnerabilities in automotive systems. Sections IV, V, VI, and VII describe various categories of research in automotive security. In Section VIII we closely examine one specific, celebrated vehicle hack, *e.g.*, by Miller and Valasek in 2015; the goal of this case study is to examine how vulnerabilities at different levels can be “chained” together by a hacker to compromise a vehicle. In Section IX we provide a high-level overview of industrial practice in identifying security vulnerabilities. Section X briefly discusses security challenges with the automotive supply chain. We conclude the paper in Section XI. Note that while this paper comprehensively discusses automotive security at the vehicle level, automotive security involves many other components, *e.g.*, validation of individual SoCs and ECUs in automobiles, the relation between security and functional safety, etc. These components are outside the scope of this paper and we refer to previously published literature [129], [127], [125], [126] for these topics.

II. BACKGROUND

A. Electronics and Software in Modern Automotive Systems

The transformation of automotive systems from a mechanical or electro-mechanical system to a chiefly electronic one arguably began with the development of engine control

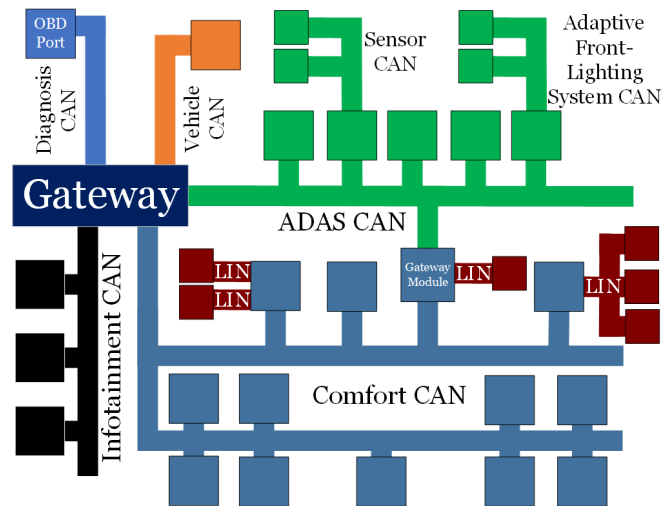


Fig. 1: Overview of an automotive system architecture. Each box refers to an ECU (controller). ECUs are connected with one another through buses and intra-networking protocols such as Controller Area Network (CAN) and Local Interconnect Network (LIN). CAN is primarily used for core driving functionality and engine control as well as for sensors, comfort, infotainment, and the Adaptive Front-Lighting System.

and fuel injection systems in the 1970s. Starting from the 1990s, the design complexity of automotive systems has been dominated by electronic parts, with more focus on software components in the last decade. Today’s cars include electronics and software for infotainment, driver assistance (ADAS), and energy efficiency (*e.g.*, emission control), to name a few. The electronic and software components in an automobile (which we will loosely refer to as electronics) are typically divided into five functional domains:

- **Telematics:** This includes the multimedia and infotainment components of the car including radio, rear-seat entertainment, and navigation systems.
- **Body:** This includes air-conditioning and climate control, the electronic dashboard, power doors, seats, windows, mirrors, lights, park distance control, etc.
- **Chassis:** This includes features such as the Antilock Braking System, Stability Control, Adaptive Cruise Control, etc.
- **Powertrain:** This includes the electronics for controlling the engine, fuel injection, transmission gear, ignition timing, etc.
- **Passive Safety:** This includes all the electronics designed to add safety mechanisms, including roll-over sensors, airbags, belt pre-tensioners, etc.

Obviously, many automotive features cross-cut a variety of functional domains. For example, many modern cars include speed-compensated volume adjustment, *i.e.*, adjustment of multimedia volume in response to increasing speed of the car. This requires communication between the radio (part of telematics) and ADAS components. Other similar examples include automatic braking while reversing if the backup camera senses a child or small obstacle and showing the reversing

trajectory on display (which requires computation of angular momentum of the wheels). To enable these features, automotive system architectures involve significant and complex communication among the different in-vehicle components. This is implemented through a variety of protocols including Controller Area Network Bus (CAN-Bus), Local Interconnect Network (LIN), FlexRay, and Media Oriented Systems Transport (MOST). Fig. 1 provides a representative automotive architecture. In addition to in-vehicle communication, current and emergent vehicles also communicate with external entities (e.g., other cars, infrastructure components, etc.).

B. Security Requirements

Traditionally, the security of a Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related (E/E/PE) System includes the following foundational pillars: confidentiality, integrity, and availability, also referred to as the *CIA pillars* [57]. More recently, authentication and repudiation have been added as additional pillars, particularly for communicating systems and devices.

- 1) **Confidentiality:** This refers to the requirement that sensitive, critical system information and data are not perceivable by parties who are not the intended recipients.
- 2) **Integrity:** This refers to the requirement that an unauthorized entity cannot corrupt or modify sensitive data or information. In the context of communicating agents, integrity involves the requirement that data received is not different than what was originally intended to be sent. Further, the data should be accompanied by a warranty that it was sent from the expected user at an expected time.
- 3) **Availability:** This refers to the requirement that a legitimate user or application can access requested resources and perform functions within a guaranteed time limit. An obvious subversion on availability is a denial-of-service (DOS) attack.
- 4) **Authentication:** The assurance that communicating parties can verify the identity of each other and that parties are only able to attain access to resources corresponding to their access level.
- 5) **Non-Repudiation:** This refers to the assurance that a party cannot refute something they have done (e.g., sending a packet). It requires a mechanism to prove the history of a communicating party. This usually involves a combination of authentication and integrity.

Of course, the above requirements are very general. Translating them for a specific application involves definitions of *security policies* targeted towards that system, e.g., confidentiality requirements are enforced through security policies that stipulate how sensitive assets in the system can be accessed and the agents and devices authorized to access them [12], [13], [129]. Nevertheless, the five pillars above can be used to systematize and categorize security attacks and defenses. For this paper, when discussing security vulnerabilities on automotive systems, we will use this taxonomy to categorize both attacks and defenses.

First Author	Year	Experimental Surface	Citation
Koscher	2010	Unknown	[87]
Checkoway	2011	Unknown	[31]
Foster	2015	Mobile Devices Engineering TCU used by Uber	[50]
Miller	2012	2010 Ford Escape & 2010 Toyota Prius	[110]
Miller	2015	2014 Jeep Cherokee	[111]
Keen Security Lab	2016-19	Tesla Model S, BMW	[155], [156], [157], [154]
Smith	2016	2006 Chevrolet Malibu	[144]

TABLE I: Summary of works that hacked on-road vehicles to study their vulnerabilities and potential exploits

III. SOME CHALLENGES WITH AUTOMOTIVE SECURITY

At a high level, security attacks on automotive systems are obvious instances of general cybersecurity problems. In particular, a large number of electronic and software components that were not originally designed to be connected to the Internet are now connecting to the Internet, so it is unsurprising that security vulnerabilities exist which can then be exploited in the field. On the other hand, one challenge is that traditional cybersecurity solutions cannot be directly used to *mitigate* such attacks. In particular, automotive systems are in the field for a long time compared to traditional information technology (IT) systems, mobile systems, and wearable devices. For instance, a mobile phone remains in the field for a couple of years. In contrast, a car may remain in the field for a decade or more. This gives the hacker a long time to find vulnerabilities in deployed vehicles. Furthermore, even if there is no security problem at the time of deployment, security requirements can change within this long life-time, adversely impacting the level of security assurance on a deployed, mature system. Furthermore, traditional security assurance solutions, e.g., encryption, authentication, etc., are typically computationally intensive. It is difficult to deploy many of these solutions with the memory and computation constraints of automotive ECU's. Finally, such solutions may raise issues related to privacy. For instance, in connected platoons, strong authentication may be desirable to ensure that a vehicle-to-vehicle communication is indeed coming from an authentic vehicle; however, a strong authentication scheme may disclose the identity of the sending vehicle, which can then be used to extract various private information including location and driving history.

IV. A SAMPLING OF AUTOMOTIVE SECURITY ATTACKS

For the last decade, researchers and white-hat hackers have been experimenting on advanced automotive systems to analyze them and discover vulnerabilities. Their primary purpose in doing this is to showcase the need for secure automotive systems as more and more capabilities are added to them over time. Each entity has used unique methods and experimental setups to conduct their studies. We will study one of these hacks in-depth in Section VIII. In this section, we provide a high-level overview of the different hacks to

give a general flavor of the spectrum of techniques involved. Table I provides a summary of these hacks and the related publications organized by author, year, experimental surface (vehicle type) and citations.

One of the earliest comprehensive attacks on an in-field automobile (they did not publish details of the vehicle involved) was performed by Koscher *et al.* [87] in 2010. The work primarily involved exploits based on physical attacks. Subsequently, two other papers were published following up on the original hack, *e.g.*, by Checkoway *et al.* [31] and Foster *et al.* [50]. Checkoway *et al.* expanded upon Koscher *et al.*'s work with remote exploits that take advantage of the vehicle's telematics system. Foster *et al.* provide a thorough vulnerability analysis of the in-vehicle systems. They considered vulnerabilities of newly introduced (at the time of publication of their work) technologies including new telematic control units (TCU) to both direct and remote attacks.

A landmark study in automotive hacking was performed in 2015 by Miller and Valasek [111] who exhibited a way to remotely control the driver assistance system of a 2014 Jeep Cherokee and drive it off the road. We will discuss their work in some detail in Section VIII. This work is particularly relevant to the research community since it provides detailed documentation and explanation to enable the reproduction of their results.

In addition to the above, there has been work by white-hat hackers and research teams to discover and perform exploits on automotive systems primarily to facilitate research and awareness. One such team: the Keen Security Lab of Tencent [158], discovered exploits on several models of the Tesla Model S, including remote attacks through the CAN module and firmware over-the-air (OTA) updates. They also performed a thorough assessment of in-vehicle equipment in several BMW vehicle models and found many vulnerabilities. Another organization known as the Car Hacking Village [94], comprising several DefCon hackers, published a detailed guide for the ethical hacking of automotive systems [144]. They also include a list of recommended equipment to use. They demonstrate their approach on a 2006 Chevrolet Malibu Sedan, although their techniques apply to other vehicles.

V. SECURITY OF IN-VEHICLE NETWORKS

The functionalities of automotive systems are typically implemented through the communication and coordination of ECUs and MCUs across several in-vehicle networks. For obvious reasons, these in-vehicle networks are the primary targets of automotive security exploitation: the goal is typically to "fool" the networks into communicating or delivering unauthorized messages. Since many of the messages carried by these networks can have significant impacts on vehicular functionality, *e.g.*, messages through the CAN network can affect vital driving functions including braking and cruise control, a successful attack on the network would typically lead to the compromise of the entire system. Table II shows the differences between several common in-vehicle network protocols.

Wolf [168] discusses many inherent vulnerabilities of in-vehicle networks. For example, LIN uses a master-slave archi-

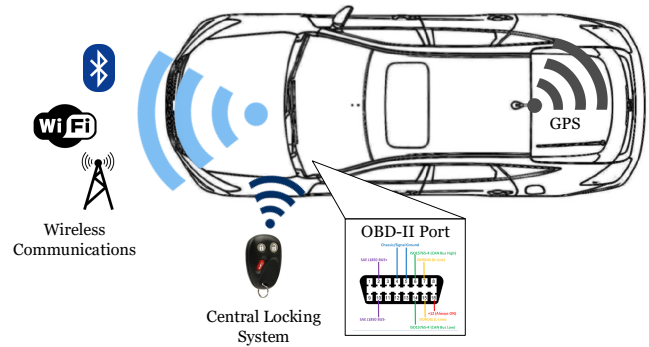


Fig. 2: Common internal and remote attack vectors on automotive software systems are the diagnostic ports and telematics system.

ture with all communication initiated by the master [133]. If the master is compromised then the entire sub-network of slave nodes can be disabled. Since LIN networks often control auxiliary features such as windows, lights, mirrors, fans, etc., this can impact vehicle usability. Another example is the MOST protocol, which is primarily used for multimedia and infotainment. MOST networks use a ring or daisy-chain topology and have a single timing master node that continuously sends timing frames to synchronize slaves [58]. Since this is the only form of synchronization in the network, an attacker can send malicious timing frames to desynchronize nodes and disable the bus. This can render infotainment/telematics devices inoperable. An even greater vulnerability exists with CAN and FlexRay networks as they are designed for high speed, real-time systems and are often implemented in safety-critical applications. Both FlexRay and CAN are susceptible to exploits due to lack of authentication or encryption [106], [145]. This can allow attackers to cause malfunctions in safety-critical systems such as stability control, anti-lock braking, and engine management as well as in drive-by-wire systems such as electronic throttle and steering. In this section, we primarily focus on the security of CAN networks, since they have been ubiquitous and often form the primary communication bus in most vehicles. Many recent studies [31], [87], [50], have shown that it can be compromised by an attacker with relative ease, allowing them to enable or disable critical safety systems.

The two most common attacks for CAN-Bus exploitation are through (1) diagnostic ports and (2) telematics or infotainment systems. Fig. 2 illustrates the different attack vectors. Diagnostic ports are a common entry point for an attacker due to the relative ease associated with launching an attack, assuming the attacker has physical access to the diagnostic port (*i.e.*, access to the vehicle). Telematics and infotainment systems often use wireless protocols such as Bluetooth, cellular 2G/3G/4G, WiFi, and GPS, which enable attackers to remotely interface with these systems and launch attacks.

Protocol	Interface	Topology	Bandwidth	Transmission	Arbitration
CAN [145]	Multi-Master	Bus	1 Mbps	Asynchronous	Bit-wise arbitration where lowest message ID gets control of the bus.
LIN [133]	Master-Slave	Bus	20 Kbps	Synchronous	All messages are initiated by the master and one or zero slaves will respond to a given message.
MOST [58]	Timing Master-Slave	Daisy Chain or Star	150 Mbps	Synchronous	Access token is passed around the bus in a circle. A node can only transmit data if it has the token, ensuring fair access for all nodes.
FlexRay [106]	Multi-Master	Star or Bus	10 Mbps	Synchronous and Asynchronous	Static segment with fixed interval messages and dynamic segment with CAN-like arbitration.

TABLE II: Comparison between several common in-vehicle network protocols.

A. Attacks through Physical Access

In the United States, all vehicles sold since 1996 are required to use an OBD-II diagnostic port (specified in SAE J1962) to transmit emissions-related codes and data for vehicular emissions testing. Additionally, US legislation requires all vehicles sold since 2008 to support the ISO 15765-defined CAN standard through this OBD-II interface. Although the requirement is only for emissions-related information, most manufacturers use it as a primary diagnostic and reprogramming port as well. Since the port directly connects to several onboard computers via CAN, an attacker with physical access to the vehicle can easily launch attacks and compromise critical vehicle systems. The attacker could be an individual with legitimate access, (*e.g.*, a valet driver or mechanic), or someone who gains illegitimate access *e.g.*, through burglary. Once the attacker gains physical access, there is a wide array of OBD-II adapters available online to allow them to transmit and receive CAN messages.

Attacks through physical access, while easy to administer, have not been perceived as a "real" threat to automotive security. A standard response to such an attack has been that, if the attacker did have physical access, they could simply cut the brake wire or perform other similar damage, rather than hacking the vehicle through a CAN network. Nevertheless, as attention to automotive security has intensified in recent years, there have been efforts to mitigate such attacks. To combat exploits that utilize the physical OBD-II port, Markham *et al.* [107] proposed a role-based access control policy: each commercial OBD-II device would be certified by manufacturers and would send a public key and X.509 certificate to the vehicle to prove its identity. Once a device is verified by the vehicle, it is given access to certain systems based on its privilege; non-certified devices would only have permission to read the bus, while a certified mechanic's scan tool would have both read and write permissions. Nevertheless, note that attacks similar to the PassThru exploit [31] would circumvent this form of authentication.

B. Remote Attacks through Infotainment/Telematics

The Defense Advanced Research Projects Agency (DARPA) has demonstrated exploitable hacks in vehicular infotainment applications such as the UConnect® system in Chrysler, Jeep, FIAT, etc. [109]. They demonstrated that they could remotely control a vehicle via CAN bus commands. Their hacking demonstrations resulted in several recalls, including 1.4 million Chrysler automobiles [56].

Since 2004, the Environmental Protection Agency requires all vehicles manufactured in the US to support SAE J2534 PassThru devices, allowing Windows computers to communicate with a vehicle's internal bus networks. Consequently, many mechanics and technicians use J2534 PassThru devices for diagnostics and emissions testing. PassThru devices connect to the OBD-II port in vehicles and communicate with the Windows machine via a wired or wireless network. Checkoway *et al.* [31] showed how it is possible to hack these devices remotely through a local WiFi network. Since the PassThru device used by Checkoway depended on external network security, its communication over the local wireless network was not secured. This allowed them to perform a shell injection and install malicious binaries on the PassThru device and use the PassThru to install malicious code on a connected vehicle as well. Checkoway also demonstrated that a worm could be implemented to copy malicious code between multiple PassThru devices on the same network, increasing the impact of this attack. Other remote attacks include the exploitation of vulnerabilities in infotainment/telematics systems. These systems often include interfaces for Bluetooth, cellular, GPS, and other wireless protocols, as well as a communication channel to the internal CAN network; this makes them particularly attractive targets for remote attacks. Exploits to these systems often involve traditional hardware and software security exploits. For example, Checkoway *et al.* [31] showed a buffer overflow attack by installing a simple Trojan application on a connected Android phone; the application listened to Bluetooth traffic to determine if a certain model of telematics unit was connected and, if so, delivered the attack payload. Furthermore, using the bridging capability of the infotainment system, they could send arbitrary CAN messages to the internal CAN network.

C. Integrity and Availability Attacks

CAN was designed for real-time systems and prioritizes speed and reliability of delivery. CAN messages are broadcast to every node in the network, permitting anyone with bus access to perform packet sniffing. Additionally, CAN messages do not contain any authentication information to verify senders; the message ID is the only identifier used by a node to determine if it should process a message. This enables attackers to easily perform replay attacks by sending packets with message IDs that match the IDs of legitimate messages they want to spoof. Since CAN messages control various driving functions, attacks on integrity and availability can be mounted through appropriate CAN messages. For

instance, Koscher *et al.* [87] showed how to send specific CAN messages in consumer vehicles that utilize electronic stability and brake control (*e.g.*, ABS braking) to enable and disable brakes at speed.

It is difficult to prevent replay and availability attacks on CAN networks without significant protocol changes. However, there has been interest in detecting attacks non-intrusively by checking for anomalous bus traffic. Several intrusion detection strategies have been developed to defend against attacks on in-vehicle systems. Taylor *et al.* [151] demonstrated a non-intrusive anomaly detector for identifying replay attacks on CAN networks. The algorithm measures inter-packet timing over a sliding window and compares average times to historical averages to create an anomaly signal, and targets both replay and availability attacks. Note, however, that while such a solution is effective at detecting availability and replay attacks for periodic messages, they are ineffective at detecting attacks involving non-periodic messages due to their reliance on historic timing averages. Additionally, since these methods do not check message data for anomalies (only message timing), an attacker who can modify data within periodic messages without affecting timing intervals would be able to subvert these methods. Cho *et al.* [35] proposed an anomaly-based intrusion detection system (IDS) that utilizes the intervals and clock skews of periodic in-vehicle messages to create unique fingerprints for each ECU. Deviations from this signature indicate an intrusion into the network by a compromised ECU or other device. The proposed IDS was able to detect these intrusions with a false positive rate of 0.055%.

D. Authentication and Non-Repudiation Attacks

CAN networks have many restrictions that make it difficult to implement many known authentication protocols. Herrewege [161] discusses many of these restrictions. First, since CAN networks often have hard real-time constraints, one cannot introduce an authentication protocol that significantly impacts message timings. Second, each CAN message can only contain a maximum of 8 bytes, meaning that extra authentication data cannot simply be appended to existing messages. Third, since CAN message IDs correspond to specific functions, it is not feasible to add extra IDs for authentication data. Finally, the unidirectional message-passing methodology used by CAN makes it difficult to directly address specific nodes without using a rudimentary method such as flags. The combination of these factors makes it difficult for vehicle manufacturers to implement secure access control policies without significant time or capital investment. Car manufacturers typically prevent unauthorized re-flashing of the software on ECUs through CAN; however, the restrictions discussed above imply that only light-weight authentications can be implemented, which can be easily bypassed resulting in integrity and non-repudiation attacks through unauthorized, over-the-air update. Koscher *et al.* [87] showed that in a mid-range consumer vehicle, the authentication scheme used to control write access is a simple challenge-response pair: the car will ask for a 16-bit key which, if provided, unlocks the ECU software. They demonstrated that this form of key can be cracked with brute force in a matter of days.

Some manufacturers choose to keep critical systems on a separate, high-speed bus instead of on the primary bus so that critical systems are not affected if the primary bus is disabled. This can help prevent attacks on critical systems; however, since it is relatively easy for an attacker to re-flash ECUs, the attacker could compromise any ECU which communicates on both networks. For example, Koscher *et al.* [87] showed that the telematics unit (which communicates with both networks) was able to be reprogrammed from the low-speed bus to send custom messages on the high-speed bus. We will discuss a more detailed effect of this exploit in Section VIII. Furthermore, Koscher *et al.* showed how to apply software reprogramming to launch non-repudiation attacks as follows. One can introduce a Trojan software by re-flashing the ECU such that the existing functionality would not be affected, allowing the original software and the malware to coexist. After the malware executes an attack (*e.g.*, disabling the engine or locking the brakes), it would delete itself and relevant log data from the ECU to prevent detection during a forensic investigation.

Addressing the above attacks requires the development of authentication protocols that can meet CAN's real-time requirements. Herrewege *et al.* [161] presented a message authentication protocol named CANAuth, which inserts a hashed message authentication code (HMAC) between sampling points of a CAN bus interface. This is done using the CAN+ protocol proposed earlier by Ziermann [180] to encode data at a higher frequency within a single CAN bit without interfering with the underlying CAN bus protocol.

E. Ransomware and Thefts through CAN

The idea of a ransomware attack is to make a system functionality inaccessible to the user and demand ransom in exchange for returning access. In cybersecurity, this attack typically takes the form of encrypting important system files or locking functionality. In automotive systems, however, an attacker with the ability to send CAN messages can easily mount ransomware attacks by gaining control over a variety of in-vehicle functionalities. For instance, most vehicles today employ a central locking system and use CAN to control in-vehicle displays and user interfaces. Koscher *et al.* [87] showed that it is easy to control the locking of vehicle doors; turning on the horn; activating and deactivating the Heating, Ventilation, and Cooling (HVAC) system; or displaying arbitrary messages through the panel cluster display. In addition to ransomware, it is easy to use CAN to enable theft of the vehicle silently without activating the alarm, *e.g.*, by sending messages to the telematics unit to successively (1) unlock the doors, (2) disable the immobilizer, and (3) start the engine.

VI. SECURITY OF VEHICULAR COMMUNICATIONS

A key feature of emergent autonomous vehicles is the ability to communicate with other vehicles, with the infrastructure, and with other devices connected to the Internet. Many features of autonomous transportation depend on such communications, including platooning, cooperative route management,

etc. Consequently, there has been significant interest in designing effective vehicular communications. In this section, we look at potential threats on such communications and the proposed defenses.

Vehicular communication or Vehicular-to-Everything (V2X) has been introduced as an amendment to the IEEE 802.11p standard. The standard was originally intended for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications [3]. However, Vehicle-to-IoT (V2IoT) communications are anticipated to be implemented and standardized in the near future [7]. 5G-based Cellular-V2X is also being introduced by companies such as Qualcomm to compete against 802.11p as the leading V2X standard. 5G promises to be revolutionary for V2X due to its higher bandwidth capacity, smaller cell sizes, and new beam-forming capabilities relative to 4G Long-Term Evolution (LTE) [18], [104]. Fig. 3 provides a visualization of the connected environment induced by V2X. A major part of security challenges in V2X is inherited from, — and similar to, — those in non-mobile ad hoc wireless networks. However, a compromise in V2X is much more serious since automobiles are safety-critical, electro-mechanical systems that influence major factors of peoples' lives [48]. For this reason, there have been efforts from standards bodies including IEEE and the European Telecommunications Standards Institute (ETSI) to develop standards and guidelines for V2X communication and Intelligent Transportation Systems (ITS) to meet the security objectives [3], [45], [167]. However, due to the complexity of these systems and their subsystems, it is challenging to guarantee or even satisfy a majority of these security objectives [135]. Attack vectors targeting V2X are large and diverse as the methods used to compromise a vehicle's security depend on the kinds of entry points accessible to the attacker. V2X attacks are generally categorized by sophistication levels based on the distance between the attacker and the target vehicle. A direct/physical attack can be mounted by an attacker who is able to obtain physical access to the vehicle or hardware (e.g., OBU, CAN bus, transceivers), either as the owner of the vehicle or via successful attacks/exploits of the CAN network as discussed in Section V. A remote attack is mounted by an attacker that does not have direct access to the vehicle.

A. Confidentiality Attacks

Confidentiality may be breached if an attacker directly accesses their OBU (as shown in the previous section) or purchases and implements 802.11p on a Software-Defined Radio (SDR) to sniff packets containing private or critical information [23]. For example, they could track a nearby vehicle via the position, speed, and action identities (unique to the event and contains information about originator) in V2V Decentralized Environmental Notification Messages (DENM) [135] or steal someone's credit card information transmitted over the air for Electronic Toll Collection (ETC) [93]. Tracking may lead to identifying a driver's behaviors, personal interests, home/work address, and/or their real identity [24], [72]. In the future, when peer-to-peer sharing is implemented, a key challenge will be to ensure the privacy of network traffic between vehicles, infrastructure, and IoT devices [96], [48].

To address the requirement for confidentiality, asymmetric key-based encryption methods have been proposed (e.g., Elliptical Curve Cryptography) by IEEE [3]. However, these methods are costly and challenging to implement in the ad hoc, heterogeneous environment of V2V communications [135]. Another challenge is latency, which must be kept to a minimum (less than or equal to 50ms for triggering events and resulting actions). Thus, ensuring that a cryptographic solution is both reliable and fast is a major challenge [135]. Furthermore, encryption methods should also be able to adapt to the situation (emergency or entertainment) to reduce energy and timing costs and ensure both safety and security. There have been proposed solutions [165], [164] which attempt to use the benefits of the dynamically-changing physical environment to quickly generate highly random, symmetric cryptographic keys by adapting to the energy and timing constraints of V2X scenarios and the reciprocal fading properties of the wireless channel. Other solutions [44], [149], [170] attempt to algorithmically reduce the overhead of existing cryptographic solutions.

To prevent attacks on privacy, researchers have recommended using pseudonyms, sending data during only a part of the taken route (rather than all), ensuring k -anonymity, and consistently updating unique identifiers such as the MAC address, public key certificates, and probe message IDs [24], [98]. Some solutions provide domain-specific mitigation and prevention approaches for specific applications such as Electronic Toll Booth Collection [11], [108].

B. Integrity, Authentication and Non-Repudiation

Attacks on integrity and authentication typically involve tampering, fuzzing, and spoofing in some fashion. Tampering or fuzzing attacks involve the modification or injection of noise into packets sent over the air to confuse the involved parties, but they do not require attackers to masquerade as others. Besides remote V2X attacks, attackers may maliciously alter the code of in-vehicle CPUs, e.g., using malware or reflashing (via physical access to OBU or remote attacks to the telematics/infotainment from V2IoT [116]), or modify the original data before it is transmitted. Spoofing attacks, such as the Sybil attack, are detrimental to network productivity and breach both the integrity and authentication security objectives.

Douceur [43] proposed the Sybil attack. It involves a malicious node that adopts multiple addresses of legitimate (Sybil) nodes. This means that all messages will be rerouted to this malicious node instead of the legitimate nodes. Having these messages, the attacker can tamper with them and resend them to the legitimate nodes, or deceive nearby vehicles into believing that they are surrounded by traffic to get an empty route for itself once others choose alternate routes [43], [89]. Another attack method: tunneling, involves imitating a short wireless channel between two legitimate nodes from both ends of a network [124]. It causes the two legitimate nodes to select the malicious node in their routing algorithms. In turn, this allows the malicious node to infer information about the nodes, modify packets, and delay their communication attack availability. Timing-faking attacks were also shown to

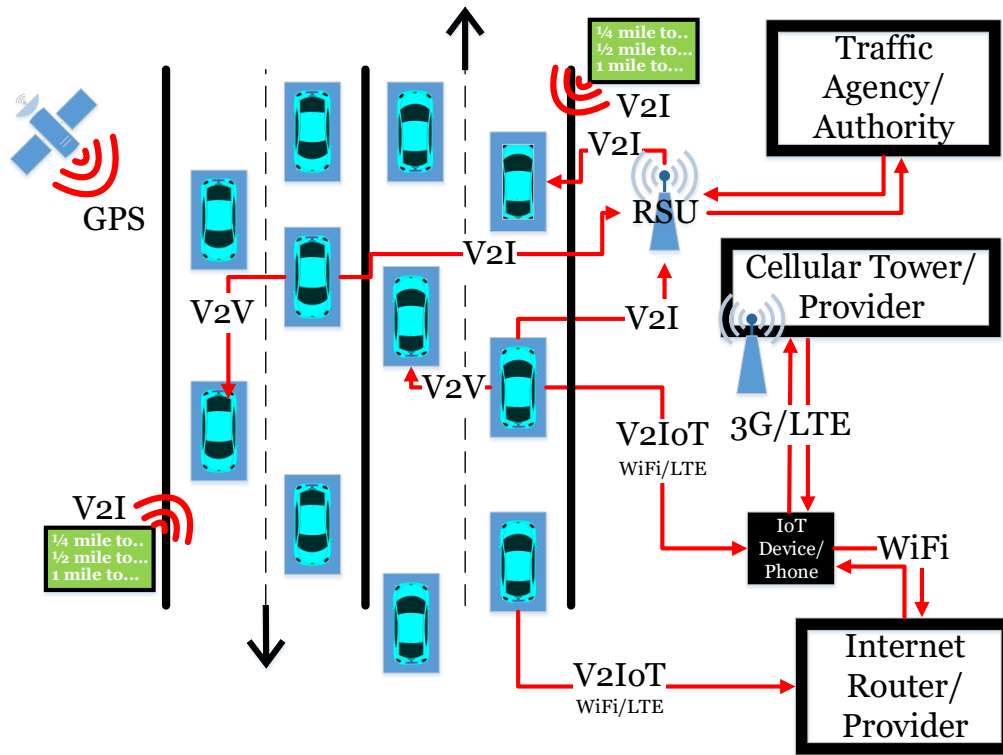


Fig. 3: Vehicular communication aka Vehicle-to-Everything (V2X) environment with traditional, connected, and autonomous vehicles. Each line corresponds to a type of one or two-way communication channel for a specific application (V2V, V2I, V2IoT). Each connectivity line may also represent a potential attack vector for an exploitation.

be effective against V2X systems. By delaying the timing of packets, RSUs will end up making incorrect decisions and force vehicles to enter sub-optimal routes with traffic, accidents, or other unforeseen circumstances [146].

Attacks on integrity and/or authentication could lead to a variety of impacts, including but not limited to traffic congestion and extra fuel costs, lower travel time for an attacker, ransomware, injury, and even murder. Garip *et al.* [52] showed how to simulate V2V attacks on connected autonomous vehicles using botnets (many bot vehicles in a targeted area). In their Manhattan grid experiments, they discovered that such attacks could overcome correlation-based defenses and cause traffic congestion (increase in average trip time by 50%) when only 1% of traffic is in the bot-net area. Various recent researches [54], [130], [131], [33] discovered that traffic controllers were highly vulnerable to spoofing attacks. These attacks will lead to sub-optimal signal timing plans at intersections or freeway ramps to cause more traffic congestion, reduced travel times for attackers, or even accidents due to spoofed or tampered traffic signal information or unexpected timing changes and distracted/unaware drivers. Wireless authentication is also being implemented with Electric Vehicles (EVs) and the Smart Grid. In particular, there is a standardization where EVs would be able to use keys and certificates to wirelessly authenticate with a charging station and recharge the vehicle [29], [30]. However, an attacker (car

thief) nearby may perform a substitution attack and use the victim's credentials instead of their own (which are invalid) to charge their vehicle.

Attacks that violate non-repudiation typically either directly target related security requirements (*i.e.*, integrity, authentication, and availability) or directly target weak points in the non-repudiation schemes. For example, an attack may involve deciphering a weak cryptographic key used in a non-repudiation scheme (*e.g.*, digital signature) or delaying mechanisms that verify the action history of a vehicle or node (*e.g.*, voting, blockchain) [162], [41], [9].

Defending against attacks on vehicular communications is of crucial importance to the proliferation of connected vehicles. In the 802.11p/WAVE standard, the necessary protection mechanisms provided for integrity include using a Message Authentication Code (MAC) (if using symmetric keys) over the data, and a digital signature (if using asymmetric keys and identities) via RSUs and/or authorities like the Department of Motor Vehicles [64], [147], [98], [122], [36], [123], [135], [8]. These solutions may ensure authentication and non-repudiation as well. Combining these with a tamper-resistant cryptographic unit (such as the Trusted Platform Module in [59]) can provide significant protection against the attacks discussed above. Maintaining timeliness and freshness additionally requires time-variant parameters [135], [8]. However, note that these defenses may push development costs

up and suffer from deterministic seeds, mismanagement of secret keys, and occasionally the tight resource constraints of embedded devices.

There has also been significant work on detection methods for integrity violations in vehicular communications. Relevant approaches include correlating messages from neighbors or using a reputation-based mechanism (via RSUs or authorities) to either infer the trustworthiness of messages or immediately detect tampered messages [21], [62], [52], [134], [27], [176]. Plausibility checks on the received data (time and location) have been proposed to prevent usage of spoofed data or even detect Sybil attacks (based on GPS data [61]). In [64], the authors propose a low-cost, position-based routing protocol using digital signatures/certificates, plausibility checks, and rate limitations to limit attacker capabilities. Another approach is to provide watchdog vehicles to monitor network traffic and identify potential attackers [67].

C. Availability

Since availability is intertwined with integrity and authentication, many of the attacks on integrity and non-repudiation discussed in Section VI-B also impact availability. Furthermore, there are many networking attacks unique to availability, *e.g.*, flooding/spamming, blackholes/greyholes/wormholes, physical layer jamming, and malware. Impacts of blackholes/greyholes/wormholes are studied in many recent papers [19], [159], [163], which demonstrate attacks resulting in the network dropping packets in flight. Flooding and spamming attacks include message-based Denial of Service [22], [65]. Basciftci *et al.* [14] demonstrated a physical layer jamming attack with SDRs from National Instruments and simulated the same jamming attack in an LTE network simulation platform to cause a performance drop of over 40% for more than 50% of users. Finally, after malware injection into a vehicle, infrastructure, or IoT device, an attacker may purposely interfere with the receptions and processing of data to reduce the operational effectiveness of a vehicle and its peers.

Given the close correspondence between availability attacks and integration/repudiation attacks, defenses against the latter also serve as defenses against the former. However, there are also specific detection and prevention methods against availability attacks. Kaur *et al.* [80] present a detection and prevention technique for wormhole attacks by forcing authenticated nodes to hash their routing-based messages and also increment the number of hops appropriately for a unique decision message. If the hops were modified by the attacker, then the hash will be different than the hashed version of the legitimate message and the malicious message will be discarded. Khatoun *et al.* [83] provide a solution to identify malicious nodes performing Black Hole attacks by aggregating and analyzing information from RSUs and vehicles to measure the reliability and reputation of nodes. Jamming attacks may be mitigated or prevented via network coding techniques [53].

D. V2X and 5G

The upcoming complex 5G ecosystem is envisioned to include autonomous and connected vehicles, drones, air traffic

control, transportation systems, health, smart factories, smart homes, smart cities, cloud-driven systems (robots and virtual reality), industrial processes and much more [101], [6], [117], [28], [103]. By 2020, it is expected that over 25 billion IoT units will be connected via various wireless and wired networking protocols of all types (automotive systems alone are expected to utilize 3G, 4G LTE, IEEE 802.11p, intra-networking, Bluetooth, and ZigBee among others) [137]. The 5G ecosystem promises exciting business opportunities, but its extreme level of inter-connectivity is also a double-edged sword and comes with risks.

Due to the inter-connectivity of various devices under various protocols, the attack surface will be ever-growing and attractive for malicious entities and also terribly difficult for businesses to manage. Attacks may start from one endpoint to another endpoint in a completely different subsystem (*e.g.*, smart home to connected autonomous vehicle). Devices and protocols that were once considered too complex for attackers to target or bother with are now more commonplace and well-understood by hackers. In 2016, there was a leap in malware attacks [166], [66] and in 2017 alone, there was a 250% increase in mobile ransomware attacks due to the rapid adoption of LTE and IoT [137]. It is clear that when devices with legacy security solutions will become connected, unless properly secured, 5G devices will become attractive targets for larger-scale attacks such as the Mirai DDoS in 2016 [152], [10], [86]. The Mirai malware enabled attackers to seek out vulnerable devices via Telnet (incidentally Telnet was found to be potentially exploitable in several of the automotive aforementioned research works [87], [31], [111]) to take control over them, to prevent users from regaining control, and to utilize them to perform large-scale DDoS attacks on Internet service provider devices at the lower-layer Internet protocols [137], [152], [15]. It would not be surprising if many ECUs in vehicular networks were exploited to become a part of large scale botnet attacks (potentially up to the Tbps traffic volume scale [152]) or even the target of DoS attacks. Further, new types of networking protocols and applications resulting from 5G (*e.g.*, Software-Defined Networks (SDN), Virtual Mobile Network Operators (VMNO), Mobile Edge Computing) reduce the gap and create softer boundaries between devices. Thus, they require new security designs and solutions to prevent access-related exploits. In particular, infotainment systems of connected and autonomous vehicles will be connected to all sorts of devices for many services (*e.g.*, entertainment [113], performance like battery management [103], and traffic control [48]). They will become attractive targets for threats such as ransomware or direct vehicle control. Finally, privacy concerns on identity tracking, behavior inferences, subscribed services, locations, and mobility patterns will rise because of the need for and capability to process massive data traffic flows through 5G [101], [15]. Such vital user information may be exploited in unethical ways or may be used in spoofing attacks.

Despite these potential security risks, the large scale, virtualization, and inherent distributive properties of future 5G networks may be also useful to eliminate threats such as DDoS attacks [4], [15]. Improvements in security techniques

like firewalls, server load balancing, and FPGA-based Flexible Traffic Acceleration [137] and employment of physical layer security [164], [165], [173] will definitely help as well. In short, to address the risks of legacy software and hardware connecting with other devices through 5G, both businesses and service providers alike must strive to design their products with security in mind from the ground up.

VII. SECURITY OF VEHICULAR COMPONENTS

In addition to the communication mechanisms (whether in-vehicle or V2X), electronics components in the vehicle are also obviously subject to attacks. In this section, we consider these attacks and their effects on vehicular security.

A. Privacy Attacks on Infotainment Components

Most modern consumer vehicles have voice control or hands-free calling to allow users to keep their eyes on the road while using infotainment systems or making phone calls. Note that the microphone remains active for the entire duration of phone calls. These technologies can be exploited by an attacker to covertly record audio inside the vehicle. Checkoway *et al.* [31] demonstrated how to use a compromised telematics unit to record audio from an in-cabin microphone and stream it through a cellular network. Furthermore, vehicle location data can be extracted from the telematics unit as well, enabling attackers to monitor a user's location at all times. This could be used to identify high-value targets, such as owners of expensive vehicles who park at large corporations, to potentially find their home address for further surveillance or theft. The attacks on aftermarket TCUs demonstrated by Foster also facilitate these forms of data extraction [50].

B. Attacks on Wireless Key Entry and Ignition

Since the mid-1990s, Radio Frequency Identification (RFID), Remote Keyless Entry (RKE) and/or Remote Keyless Ignition (RKI) have commonly been implemented for consumer comfort and vehicle security against thieves. Ironically, these wireless communication-based solutions are also insecure. There have been several works [25], [144], [51] that found vulnerabilities in all three applications primarily because of design errors. Furthermore, stringent cost requirements make the implementation of advanced and sophisticated protection in these areas challenging. In particular, signals to open or lock a car or start the engine could be stored, blocked, or relayed either wirelessly or over a cable. Such attacks could allow thieves to unlock and/or start a vehicle despite its owner being physically away. In 2005, a Texas Instrument RFID transponder used as an ignition key in millions of vehicles was also found to be hackable due to weak cryptographic keys [25]. In 2015, Kamkar developed and presented the RollJam attack on RKE [79]. The RollJam exploit simultaneously stores and jams a signal sent to unlock the door; then, when the driver sends an unlock signal to the door again, it is again blocked but the first stored signal is sent instead to the vehicle's receiver. This allows the attacker to use the second stored signal to unlock the car at will. Such an attack would

only cost approximately \$32 and it was successfully tested on Nissan, Cadillac, Ford, Toyota, Lotus, Volkswagen, and Chrysler vehicles. Ibrahim *et al.* [73] demonstrated a three-step attack involving (1) set-up, (2) jamming and recording, and (3) hijacking. Their attacks were more flexible than the RollJam attack (remotely controller jammer, no need for precisely tuned jammer) and easier (constant jamming forces user to eventually use mechanical key and not reset the RKE code). They tested their attack with various distance parameters (distance from user to vehicle and distance from user to attacker's logger device) on six vehicles Škoda Yeti (2016), Škoda Octavia (2009), Mazda 6 (2009), Toyota Rav4 (2014), Mitsubishi Pajero (2015) and Nissan Sunny (2014).

Note that the challenges to securing wireless key entry and ignition systems include resource limitations of hardware and the high costs of cryptographic solutions. Most researcher recommendations include using RF signal properties to verify if a user is truly nearby or not [84], [115], [136]. Yang *et al.* [174] propose a low cost (memory and complexity) solution that involves a challenge-response protocol based on distance bounding, where the verifier measures an upper-bound of the actual distance to the prover so that the attacker cannot convince them that they are closer than they really are. Furthermore, a possible solution to wireless attacks on wireless authentication between EVs and the smart grid has been recommended through a "cyber-physical authentication protocol" which requires physical access of the charging cable to verify the identity and legitimacy of a vehicle [30].

C. Sensor Attacks

Integrated and embedded sensors in automotive systems are crucial for the operation of connected and autonomous vehicles. With wireless communication, connected vehicles can exchange sensor data with each other for smarter applications and better control. Vehicles with autonomous capability need more informative and accurate sensors (*e.g.*, LIDAR and camera), and more efficient and reliable algorithms for control (*e.g.*, machine learning models) [97], [72]. Consequently, the security of sensors is critical to prevent severe functional and safety-critical impacts from exploits. Unfortunately, these sensors and sensor data-based algorithms are heavily vulnerable to malicious environmental and wireless communication modifications.

Rouf *et al.* developed an attack with a low-cost software-defined radio (SDR) that captured and read Tire Pressure Monitoring System (TPMS) communication packets from a vehicle up to 40 meters away [75]. TPMS messages also include identifiers of tire sensors that are sufficiently unique for attackers to use to track the vehicle. Furthermore, they demonstrated the possibility of injecting packets into the TPMS network to trigger a fake warning signal [75]. Several of the hacking works mentioned in Table I have experimented and demonstrated TPMS remote attacks on their testbeds. There have also been studies of attacks on navigation systems [71], [70], [82]. These studies found that the GPS receiver was vulnerable to spoofing. Spoofing attacks would provide false location information and may lead to

longer trip times or, worse, accidents. Correspondingly, radar, another sensory component used to measure distances, is also susceptible to jamming and spoofing [172]. Furthermore, recent research [120], [63] discovered that lasers or similar technology could spoof the existence of vehicles to LIDAR. Image-based machine learning algorithms and models can also be fooled to make incorrect and life-endangering decisions if small modifications were made to road signs or lines (e.g., stickers, markings, delineation) [172], [120], [40], [5], [37], [121], [143], [154], [69].

In addition to the above, given the complexity of autonomous and connected vehicles, there is a strong necessity to have miscellaneous sensors everywhere to ensure safety and performance. Examples include gyroscopes and anti-lock braking system (ABS) sensors as well as visible light, infrared, thermal infrared, odometric (accelerometers, gyroscopes, etc.), and acoustic sensors. Attacks on these types of sensors vary in difficulty because of their varying accessibility levels [119], [72]. ABS sensors could be spoofed or jammed via an electromagnetic actuator that can be as far as three meters away from the wheel speed sensors [138]. Visible light, infrared, and thermal infrared sensors can all be deceived or jammed with environment-based injections of the same medium (but attacks may be difficult due to limited ranges) [119], [132]. Magnetic or thermal attacks may potentially affect odometric sensors to affect vehicle navigation but the success probability is low due to the hardware costs, range, and timing of such attacks.

For attacks that attempt to deceive the sensor-based algorithms with changes in the environment, Yan *et al.* [172] applied redundancy, logic checking, confidence priority, and attack detection along with sensor fusion. Petit *et al.* [120] applied low-cost software solutions such as random sampling, multiple sampling (for LIDAR), and a shortened pulse period. For eavesdropping and data spoofing attacks on network-based sensors such as TPMS, typical mitigation approaches include better logic consistency checks and low-cost cryptographic solutions with freshness and a strong source of randomness to prevent unauthorized access or usage of fake data [135], [75]. GPS and GNSS spoofing may be prevented with Navigation Message Authentication (NMA) and replay/spoofing detection methods [70]. Assuming a multi-antenna array is being used, Danesmand *et al.* proposed a low-cost method that first detects spoofing signals, maximizes authentic signals and then attenuates the spoofing signals [39]. Another approach to detect and defend against deception-based attacks on sensors and their algorithms is to perform a design-time and runtime secure state estimation and identify which sensors are trustworthy through satisfiability solving [140].

D. Attacks on the Battery Subsystem

In modern combustion vehicles, the battery subsystem has evolved from being a simple lead-acid battery powering the vehicle electronics to a complex system that manages various energy-related demands such as engine start/stop technology, hybrid drivetrains, regenerative braking, etc. These modern battery subsystems are generally connected with the CAN-bus network. In Electric Vehicles (EVs), the battery and its

subsystem take an even more crucial role. Most EVs use large lithium-ion battery packs due to their high energy density and power output. Consequentially, lithium battery technology is extremely volatile and requires constant monitoring during charging and discharging to prevent thermal runaway (a positive feedback loop between cell temperature and internal heat generation) resulting in fire and/or an explosion. EV manufacturers struggle to meet the demand for higher battery capacity without compromising safety. This can lead to mysterious catastrophic battery failures such as the reports in 2019 of several parked Tesla vehicles catching on fire for no apparent reason [85]. EVs can also potentially present greater risks than combustion vehicles during accidents as shown by NHTSA crash tests in 2011, in which the Chevrolet Volt EV caught on fire twice, prompting the NHTSA to open an investigation into the vehicle's fire risk [42]. Furthermore, the widespread adoption of rapid charging technology (such as Tesla Superchargers) places even greater stress on EV batteries, requiring active cooling, dynamic charge rate based on cell temperature, and robust cell-balancing to ensure safe rapid charging. However, in the case that these safety measures fail, the results can be disastrous. For example, in two separate incidents in 2016 and 2019, a Tesla vehicle caught fire while plugged into a Supercharger station [91], [92]. Both fires were attributed to short circuits in the vehicles' electrical systems.

The aforementioned examples demonstrate that EV battery technology's complex architecture and high risk-factor present a large attack surface. The high impact of battery failures makes this subsystem an attractive target for attackers and presents a significant risk to EV owners. Since the battery management system is usually connected to the CAN-bus and several groups have already demonstrated cyber-physical attacks on EVs [155], [156], [69], [154], [81], it is only a matter of time before exploits targeting vulnerabilities of the battery subsystem are revealed.

Lithium-ion batteries have various failure modes ranging from reduced battery life/performance to complete battery failure and thermal runaway. The former failure modes can be triggered via excessive cell cycling, charging past 100% capacity, or malicious tampering of vehicle loads such as manipulating HVAC settings, disabling regenerative braking, or disabling the discharge limiter to deep discharge the battery. Complex, new EV control systems that use machine learning and artificial intelligence to improve efficiency such as that proposed by Lin *et al.* [99] are potential attack vectors to manipulate the battery subsystem and drivetrain of the vehicle. Several groups have shown that machine learning models are highly vulnerable to adversarial attacks [88], [46], meaning that machine learning-based control systems can potentially be leveraged to attack the battery subsystem and result in these failure modes. Thermal runaway can be induced via a combination of factors including a high charging rate, poor cooling system performance, and internal or external short circuits. This failure mode is most likely to occur with rapid-charging devices as the high power output of the chargers increases battery cell temperatures significantly and requires complex thermal management in the vehicle. Some fast-charging systems require vehicles to run active

cooling systems while charging to ensure battery temperatures do not reach critical levels. Despite these safety measures, the commands controlling charge rate and active cooling are usually sent via in-vehicle networks, such as CAN. In Section V, we showed the various ways in which attackers can gain access to in-vehicle networks; in this scenario, an attacker with access to the network could potentially manipulate bus traffic to induce thermal runaway. Although many battery subsystems have physical safety measures to prevent thermal runaway such as thermally-triggered fuses, these measures are usually irreversible, meaning attacks that induce these conditions can cause permanent damage to a battery pack and compromise its availability.

In addition to vehicle battery packs, battery subsystems are prevalent in sensors, mobile devices, and future V2IoT devices. In general, battery subsystems consist of three layers: application, battery management, and physical. Per each layer, the attack vectors used to gain access may vary and actual exploits may target the confidentiality, integrity/authentication, and availability of the battery subsystem and/or other connected subsystems [102]. Due to the need for low-cost production, battery subsystems tend to be lacking in security across all three layers [38].

At the application layer, attack vectors for battery subsystem attacks involve wireless communication (*e.g.*, vehicular, remote battery management [150]), sensors, telematics, infotainment, EV charging station cables, wireless charging [105], and in-vehicle network ports. The primary attack vector for the battery management and physical layers is the automotive battery supply chain, which consists of many steps that are prone to various exploits (*e.g.*, manufacturing, transportation, swapping, recycling). Vulnerabilities in the latter layers include weak software security/hardware, leading to access to the battery management software or the battery circuit. Attacks on confidentiality (via probes or in-vehicle network-based attacks) typically record data related to battery usage to infer user behavior patterns or user location information. Integrity/authentication exploits utilize attack vectors, such as the CAN-bus, to modify charging/discharging protocols (*e.g.*, replay, spoofing, message tampering, battery circuit tampering) to disturb battery functionality and/or the functionality of battery-dependent components [60]. Finally, availability exploits (via network-based attacks or battery circuit tampering) attempt to reduce or cut off energy provision to the components needing it [77], [49].

E. Attacks on Map-Based Navigation

Map-based navigation is an application that both connected and autonomous vehicles may utilize to reduce trip time and improve passenger comfort. Map data may be stored already in the vehicle, received from RSUs, or received from cloud-based and mobile-based applications such as Waze and Google Maps [76]. Attacks that poison these maps in storage or deceive navigation applications with ghost cars may lead to less effective navigation by vehicles and eventually, traffic congestion [175], [139].

A summary of the notable works mentioned in Sections V-VII is summarized in Table III for works on Attack Methods and in Table IV for works on Defense Methods.

VIII. DIGGING DEEPER: A CAR HACKING CASE STUDY

The preceding sections attempted to provide a structure and taxonomy to the diversity of attack surfaces on current and emergent automotive systems, and the corresponding defenses. Nevertheless, successful attacks demonstrated on automotive systems actually cross-cut many of these structures. In this section, we delve deeper into a specific, demonstrated attack on a modern automobile, *viz.*, Miller and Valasek's 2015 exploitation of a Jeep Grand Cherokee. The authors have described this attack in detail in a white paper [111], enabling researchers to identify the various vulnerabilities exploited to successfully compromise a deployed vehicle and get control over its functionality. These details make this work a good target for a pedagogical case study. In this section, we discuss the key elements of this attack and some of the high-level insights. Readers interested in further understanding are referred to their white paper.

The Miller-Valasek work was done in the backdrop of two previous works, by Koscher *et al.* [87] and Checkoway *et al.* [31]. Koscher *et al.*'s work showed that once an attacker can send CAN messages, they can easily control driving functionality; however, no means were provided for getting access to CAN messages remotely. Of course, an attacker with physical access to a victim's car can cause physical damage in other forms (*e.g.*, by cutting the brake wire). Consequently, while this demonstration was interesting, it was less than compelling. Checkoway *et al.* reported the ability to get remote access to CAN, but no details were provided. Miller and Valasek's paper described the first compelling remote exploitation on a deployed vehicle with sufficient detail for the attack to be reproducible.

The hack proceeds in three key stages: (1) compromising the head unit, (2) identifying a pathway for access to CAN from the head unit, (3) message injection into CAN to compromise driving functionality. Note that each stage is non-trivial, *e.g.* CAN message injection requires significant analysis of CAN messages.

Compromising the Head Unit: The key idea behind this attack is to exploit a vulnerability in inter-process communication (IPC). IPC is a standard means for software processes to communicate with each other, either through standardized or proprietary protocols: the typical approach is for IPC services to be implemented through software *daemons* that use a variety of "sockets" to enable communication among processes. The IPC daemon in the Grand Cherokee was a standard daemon called D-Bus, which is a highly configurable IPC daemon used in a variety of embedded systems. Typically, communication through D-Bus requires authentication. The vulnerability exploited was an open, unmonitored port in D-Bus, that enabled anonymous access. Consequently, any entity or process that could connect to that port would be provided access to the D-bus services. In particular, if a hacker could get into the wireless network of a vehicle, then, given the

Year	Application	Attack Method	Attack Impact	Complexity	Citations
2010	In-Vehicle	Physical Access and Code Exploits	Vehicle Control	Low-High	[87]
2011	In-Vehicle	Remote Access and Code Exploits	Vehicle Control	Low-High	[25], [79], [73], [31]
2011	V2V / V2I	Timing Faking Attack	Suboptimal Routes/Traffic Congestion/Accidents	Low	[146]
2012	V2X	DDOS Blackhole Attack by Synchronization	Slightly Reduced Network Performance	Low-Medium	[22]
2015	V2V	Botnet-based Spoofing Attack	Suboptimal Routes/Traffic Congestion	Medium	[52]
2015	V2X	Greyhole Attack	Extremely Reduced Network Performance	Low	[163]
2015	V2X	Physical Layer Jamming	Slightly-Extremely Reduced Network Performance	Medium-High	[14]
2015	Autonomous Navigation	Camera and LIDAR Deception	Slightly-Extreme Reduced Network Performance	Low-High	[120], [121], [175]
2018	V2I and ITS	Data Spoofing to Traffic Controller or Traffic Sensors	Traffic Congestion	Medium	[33], [54], [130], [131], [175]
2018	Autonomous Navigation	Deception with Toxic Signs	Control and Performance Loss, Life-Endangering Situations	Medium	[169], [37], [143], [5], [121]
2008, 2015, 2018	Autonomous Navigation / Route Adaptation	GPS Spoofing	Control and Performance Loss, Life-Endangering Situations	Medium	[71], [68], [177]

TABLE III: Notable works according to targeted applications, attack methods, descriptions, and complexity. We subjectively define complexity based on the requirements to launch the attack: deployment and resources (time, memory, space). There are three levels (low, medium, high), where one or more levels (ranges) are provided per attack category.

Application	Defense	Description	Complexity	Citations
V2X	Voting Architecture	Multi Agent-based voting	Medium-High	[41]
V2X	Blockchain Architecture	Verification of Authenticity, Integrity and Non-Repudiation	High	[27], [141], [171]
V2X	Certificate Management, Digital Signatures, and Message Authentication Codes	Verification of Authenticity and Integrity of Data	Medium-High	[147], [74], [98], [135], [8]
V2X	Low Cost Cryptographic Schemes	Physical Layer Key Generation and Exchange, ID-Based Cryptosystem	Low	[135], [165], [164], [36]
V2X	Watchdogs and Data-Based Malicious Behavior Detection	Spoofing/Sybil/Black-Hole/Gray-Hole Detection, Position-based Routing	Medium-High	[32], [118], [55], [62], [67], [179], [176], [134], [80], [83], [64]
V2X	Location Privacy Preservation and Pseudonyms	Pseudonyms based on ID and Context	Medium-High	[11], [20], [108], [114], [47]
V2X	Network Coding	Jamming Mitigation or Prevention	Medium-High	[53]
V2X	Vehicular Visible Light Communication, Infrared, Radar	Reduced range of connection	Medium-High	[160], [100], [142]
In-Vehicle and V2X	Malware Detection	Cloud-based On-Board Malware Defense Manager	High	[178]
In-Vehicle	Intrusion Detection	Timing-Based CAN-Bus Anomaly Detector	Low	[151]
In-Vehicle	Intrusion Detection	Clock-Based ECU Fingerprinting	Low	[35]
In-Vehicle	Role-Based Access Control	Authentication of Physical OBD-II Devices with X.509 Certificates	Medium-High	[107]
In-Vehicle	Authentication and Access Control	Encoding HMAC on top of Existing CAN-Bus Messages	Medium-High	[161]

TABLE IV: Notable works on defenses, their descriptions, and complexity. We subjectively define complexity based on the requirements to launch the defense: deployment and resources (time, memory, space). There are three levels (low, medium, high), where one or more levels (ranges) are provided per defense category.

knowledge of the port number for the specific open port, they could anonymously connect to D-Bus without requiring further authentication.

Getting Into the Vehicle Network: Obviously, the network of a car would not be open to public access, — they are typically protected by a firewall. One option is for the hacker to physically hack into a connected electronic component inside the vehicle that is connected to the car’s network; however, that would require physical access to the car. This problem was circumvented by exploiting another “feature” of the Jeep Grand Cherokee: the ability to connect from any device subscribed to the network of the car’s wireless carrier. In particular, the network carrier for the cellular modem in the Jeep’s head unit was Sprint, and this carrier provided a feature that enabled any Sprint device to communicate with any other Sprint device through the Sprint™ wireless network. Consequently, it was possible to get a Sprint burner phone, tether it to any computer, and thereby give that computer access to the Sprint network where the address and port of the victim D-Bus daemon was visible.

CAN Message Injection: Given the above steps, it became possible to remotely compromise the head unit. This enabled the attacker to have full control over the in-car infotainment including radio volume, temperature control, and the heads-up display among others. However, there was no direct connection between the head unit and the driving functionality of the car.¹ Achieving that would require the ability to inject arbitrary messages on the CAN-C bus that controlled the various ADAS components. Obviously, the head unit components were not directly connected to this bus. However, they could not be physically isolated either, since many features in the car require communication between ADAS and infotainment, *e.g.*, the ability to see the trajectory in the display while reversing, — a feature available in most modern cars, — would require communication of the angular momentum information from the wheels to the display component. To address this, the head unit includes two integrated circuits, — an ARM™ and a V850™ with different components connected. The ARM component to which the radio was connected was not permitted to send CAN messages; the V850 could send CAN messages but was not directly connected to outside connections (and consequently, compromised head unit components). However, they were connected through an SPI link that enables communication between the two processors and, furthermore, the ARM processor could reprogram the V850 system through software. This enables the hacker to use a compromised ARM processor (through exploitation of the head unit) to reprogram the V850 to accept any command provided through the SPI link. Consequently, any subsequent communication from ARM (*e.g.*, CAN messages representing directives to brake, steering, accelerator, etc.) would be accepted by V850 and passed on through the CAN-C network to the appropriate component, completing the compromise.

¹Through the control of the infotainment, they could show a speed on the display that was different from the actual speed of the car, but they could not actually affect the speed of the car.

We should add that this compromise is not possible on today’s on-road vehicles (which have been patched by Jeep and Sprint), *e.g.*, Sprint has blocked all the traffic to the exposed D-bus port thereby preventing the attacker from gaining access to the head unit of the vehicle over the internet. Nevertheless, it is worth noting the cost: Jeep had to recall over 1.4 million vehicles in response to the hack. From the perspective of this paper, it is also important to realize that a practical hack of a car actually cuts across the taxonomy we developed and typically involves multiple compromises. Nevertheless, it is depressing that it is reasonably easy to perform such a hack on a deployed automotive system.

IX. AUTOMOTIVE SECURITY VALIDATION IN PRACTICE

Given the plethora of attacks as discussed above, how does the automotive industry approach the security validation of current (and emergent) vehicles? Unfortunately, the state of the art today is primarily manual. In particular, much of the practice is based on *penetration testing* [129], *i.e.*, performing security attacks on the car in a controlled environment. The Miller-Valasek hack discussed in Section VIII provides a blueprint of the approach that can be taken to approach this complex task. More generally, penetration testing for an automotive system typically involves three steps, *e.g.*, finding an entry point, exploring and reverse-engineering various firmware code installed in the system, and finally identifying vulnerabilities in the firmware to gain control over the vehicle functionality. In this section, we provide a brief insight into the process.

A. Finding Entry Points in a Vehicle

Physical access to a car can provide a multitude of entry points for a security hacker, *e.g.*, through access to CAN via an OBD-II connector. However, as discussed in the preceding sections, it is possible to get remote access to the car. In fact, every external input to the car is typically explored as a potential entry point. In particular, most modern cars connect to the Internet through a device with cellular connectivity, such as a mobile phone. If this connection is through another device, *e.g.*, via tethering with a mobile phone, that device becomes a point of vulnerability. If the car connects directly, *e.g.*, through a cellular chip, it is more complex to find an entry point. One potential area of exploration include possible open ports that are sometimes accessible through the Internet. Another area is the variety of security certificates, *e.g.*, if the car connects through a secure socket layer (SSL), then that includes a variety of certificates, many of which include several configurations which can lead to the possibility of a misconfigured certificate. A third possibility is through a variety of remote commands. In particular, various vehicle functionalities can be accessed through mobile apps, *e.g.*, remote engine start, remote door lock/unlock, remote climate control, etc. These commands typically use a middleware service *e.g.*, an MQTT broker [2], which can provide an entry point for an attack as well.

The above techniques provide some obvious “low-hanging fruits” for penetration testing, and are often successful. However, the Internet connectivity of most vehicles is generally

more secure. In particular, most electronic components of a car are typically protected by a firewall and not accessible externally through the Internet. Accessing such components requires first getting access to a computer within the vehicle's network. The Miller-Valasek work showed one way to do this, *e.g.*, through the "feature" provided by the network provider Sprint that enabled any Sprint device to communicate with any other device within the Spring network, including devices that were within a vehicle. One avenue is to hack into one device of a car (possibly locally) and use the hacked device to enable access to other devices connected to the same network. Such attacks can be thwarted by not permitting devices in one vehicle to access devices in other vehicles even within the same network. When this is implemented, various other techniques can be used, *e.g.*, by implementing a cellular tower simulator/emulator, or applying fuzzing techniques [148] on the various external-facing software including the Bluetooth stack, USB stack, etc. In addition, many vehicles have a variety of exposed hardware interfaces including debug interfaces (*e.g.*, JTAG), serial consoles, etc. Serial consoles are used during the development phase but sometimes inadvertently left open at deployment, sometimes including shells with root privilege. Finally, one can find entry points by observing or injecting CAN messages, *e.g.*, sometimes it is possible to reprogram a CPU through CAN messages as demonstrated in Miller-Valasek.

B. Obtaining and Reverse-engineering Firmware

Simply finding an entry point is not sufficient to compromise a vehicle: one must also find ways to modify its functionality. This is typically performed through reverse-engineering and modifying the firmware. Sometimes the original firmware binary is directly available from the manufacturer's Web site or through an insider in the dealership. If not, the firmware can sometimes be lifted directly from the flash memory, or a root shell in the serial console. Once the firmware is obtained, various standard reverse-engineering tools can be applied to interpret the firmware, *e.g.*, Binwalk [78], Ida Pro [1], etc. Note that this is not a trivial matter, *e.g.*, firmware is not structured, and is often targeted for a variety of non-standard instruction set architectures (*e.g.*, V850) with complex memory layout. Furthermore, they are large, *e.g.*, of the order of Gigabytes in many cases. So it is not feasible to comprehend the entire functionality. However, it is often possible to short-change the process by identifying and interpreting specific functions or symbols. In particular, strings and symbols are sometimes left in the firmware which can provide convenient starting points for reverse-engineering.

C. Privilege Escalation

The final step in a successful automotive hack is the ability to run arbitrary code or send arbitrary messages. In traditional computing systems, most software processes do not have this privilege. However, in embedded devices, it is often common for all software processes to run with administrator privilege. Since much of automotive software has been derived from embedded systems both in design philosophy and in

implementation, this feature is often available there as well. Even if all processes do not have administrator privilege, most boot-up processes do and are obvious targets for a hack. Other processes to target are IPC daemons, *e.g.*, DBus, as demonstrated by Miller-Valasek. Finally, most automotive systems include processes for over-the-air firmware upgrades. These processes of necessity execute with administrator privilege and also have significant ability to control and modify the installed firmware; if such a process can be compromised, the hacker can exert significant control over the entire vehicular firmware.

D. Applicability of Formal Methods

Clearly, the above approaches are purely manual attacks, depending on deep human insight. It is certainly reasonable to ask why there are no systematic approaches to perform such hacks in today's practice. The answer to that question is complex, relating to unavailability and limited scalability of tools, and difficulty to integrate the tools into the complex security validation methodology. To illustrate this point, we take the example of *one* promising approach, *e.g.*, formal methods. Formal methods entail the use of mathematical reasoning to identify errors or vulnerabilities in design. In principle, it is very attractive since unlike the approaches discussed above it can provide a mathematical guarantee on the robustness of a system component or find corner-case vulnerabilities that are difficult to exercise otherwise. However, while there has been significant work on the use of formal methods for hardware and system security [129], [34], the application on automotive system-level security verification in practice is limited. There are several reasons for this. First, while there is some functional specification, a comprehensive specification of a vehicle functionality at the level of detail necessary for the applicability of formal methods is lacking. Second, even if available, such a specification would be extremely complex: even more pertinent, the implementation of an overall automotive system is extremely complicated with several cross-cutting modules related to hardware, software, and communication, together with both digital and analog/mixed-signal components. Automated formal methods such as model checking, has been used in some targeted applications for verifying functional safety in individual automotive parts, *e.g.*, individual SoC designs within an ECU [34], but it is difficult to scale such approaches beyond that level. Furthermore, a significant amount of collateral is missing at design time, *e.g.*, fuse configurations necessary for ensuring life cycle isolation are not available during the RTL design of automotive hardware where formal methods could be applicable.

The above is not to discourage the extremely important role formal methods can play in security. In fact, a greater application would only be welcome. However, for any methodology, it is important to understand its shortcomings in order to find potential areas for improvement. Perhaps one way for formal methods to become applicable is through a better design management process, *e.g.*, by ensuring all specification and other validation collateral are available at the time necessary, and models are available at different levels of abstraction (perhaps through automatic abstraction) to enable applicability of formal methods at the vehicle level.

X. SECURITY OF AUTOMOTIVE SUPPLY CHAIN

The discussion in this paper focused on automotive security at the vehicle level. Another component of automotive security entails vulnerabilities in the complex, rich supply chain involved in automotive electronics production. The supply chain of automotive electronics is complex, with several potential vulnerabilities. In this section, we briefly give a flavor of the challenges involved for the sake of completeness of the presentation. For further details, the reader is referred to a recent paper [128] that exclusively addresses this subject.

Consider an electronic part developed for an automotive system. Typically, it would be developed by some electronic part vendor, and go through several tiers of part suppliers, eventually to an automotive manufacturer who would integrate the part into an automobile. Each player in this process can introduce several sensitive assets to the part. These include cryptographic keys, Digital Rights Management (DRM) keys (for infotainment parts), proprietary firmware, etc. [129]. Furthermore, each player in the supply chain can also include rogue or malicious agents, *e.g.*, a rogue employee, a malicious CAD tool, or even an untrusted foundry. It is critical to ensure that the system is robust against attacks by such players.

Supply chain security has been an active topic of research in the hardware security community, with several excellent treatises [153], [16], [112], [17]. The security threats considered in the literature include Trojan insertions, IP piracy, cloning, counterfeit ICs, and overproduction, among others. All of these threats carry over to the automotive systems as well. However, automotive systems carry their own supply chain challenges. In particular, assets should be protected, not only after the system is in-field but also when it is with the subsequent players in the supply chain.

- 1) Assets introduced by the vendor should not be accessible to suppliers, automotive manufacturers, or end-users.
- 2) Assets introduced by a supplier or automotive manufacturer should not be accessible to any other party, including the original vendor.
- 3) All assets should be protected against side-channel attacks (*e.g.*, voltage, temperature, or clock glitch attacks).
- 4) Customer and third-party software should be protected against unauthorized access.

To exacerbate the problem, note that a part may return to the original part vendor, *e.g.*, after a field return. At this point, the part includes assets from all *subsequent* players in the supply chain which must be protected from the vendor. Additional sources of complexity arise from the fact that assets may be sprinkled across different parts, and cross-cut hardware, firmware, and software; furthermore, calls are not statically provisioned but may be created on the fly as the system executes. Finally, note that test and debug interfaces add to the vulnerabilities. These interfaces provide the user with structured access to internal architectural and design features (*e.g.*, scan chain, various design-for-debug features, etc.) for functional verification, manufacturing tests, and other related activities. Since these activities entail observability of internal states of the design (and consequently of assets stored), a key challenge is to ensure the testability of the

part while preventing unauthorized access to assets. Finally, note that once a part moves from one player to another in the supply chain (referred to as a change in the “life cycle”), asset protection is adjusted through a configuration of fuses. On the other hand, fuse programming is performed through the use of the debug interface. This creates a circular dependency between the ability to program through this interface and the effect of the programming on the interface (*e.g.*, a life cycle change may change the way the debug interface is accessed).

Addressing the problems above in today’s practice is primarily manual, based on the expertise of experienced architects and designers. However, with the growing complexity of automotive electronics, this practice is getting increasingly difficult to implement, with numerous bugs and vulnerabilities found late in production or even after deployment. Currently, the industry is exploring trust provisioning schemes to address this problem at the architecture level: the idea in which assets are provisioned by various stake-holders through a specific, centralized trust model. The trust model is typically defined by the supplier of the part who is also responsible for the architecture that enables various stake-holders to insert assets at different life cycle stages. The provisioning mechanism guarantees that (1) a service that does not need an asset does not get access to it, and (2) access and update to each asset satisfies the trust model. However, the approach is in infancy and its viability is not currently well understood. Furthermore, validation schemes have been explored to validate fuse configurations to ensure life cycle isolation, *e.g.*, through formal methods.

XI. CONCLUSION

We have provided a summary of security challenges in current and emergent vehicles. The area is vast, and our goal has been to provide a structure to the plethora of attacks that we know of today and the defenses that have been considered for such attacks. We hope this paper will be a useful starting point for researchers and practitioners in the area.

Of course, the attack surface will increase significantly as we move towards self-driving, autonomous vehicles. Nevertheless, they are coming soon, *e.g.*, major companies such as Waymo and Uber are moving towards launching close to 100,000 self-driving cars with Level 4 automation in various urban cities by 2021-22. The test vehicles are being deployed in different scenarios and have driven several millions of miles to cover the corner cases [95]. They will include advanced sensors such as Lidar, radar, etc. and multiple cameras. The computational resources needed for a self-driving car are also significantly higher to accommodate the processing of large amounts of sensor data. This, in turn, necessitates modifications to the conventional firmware and software running on the underlying ECUs. Additionally, there will be a paradigm shift from cars being viewed only as production vehicles to widespread use as autonomous ride-sharing vehicles [90]. We are not even aware of how all these issues will affect the security of these vehicles. Furthermore, the increased connectivity of vehicles and infrastructure will broaden the potential attack surface significantly. The implementation of

5G and 802.11p connectivity enables unforeseen adversarial attacks not just on vehicles, but on infrastructure, traffic, and any Internet-connected systems. The security of V2X communication has not been studied in-depth, and regrettably, the industry currently does not consider security a priority for V2X development. Since V2X technology is in its infancy, it would be prudent for researchers to focus on methods to verify the security of vehicular communication systems before they become ubiquitous.

In spite of the above, it is not all “doom and gloom”. There are certain unique characteristic features of a ride-sharing autonomous vehicle that are inherently beneficial for security [26]. For instance, the communication modules are highly customized to be locked down as there is no requirement for providing different user interfaces through Bluetooth, Web browsers, etc., unlike the production vehicles. In particular, the head unit comprising of the telematics is no longer required in a ride-sharing vehicle, and the OBD-II port can also be locked in a non-standard location. Finally, there is a certain amount of obscurity in the architecture which makes it difficult for the user to gain significant access to the vehicle, extract the firmware, and test for exploits. Nevertheless, the security of these vehicles is a critical threat and it is crucial to comprehend, analyze, and mitigate security challenges in such vehicles.

REFERENCES

- [1] “Applied Reverse Engineering with IDA Pro.” [Online]. Available: <https://resources.infosecinstitute.com/applied-reverse-engineering-ida-pro>
- [2] “MQTT.” [Online]. Available: <http://mqtt.org>
- [3] “IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments,” *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1–51, July 2010.
- [4] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, “5g security: Analysis of threats and solutions,” in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2017, pp. 193–199.
- [5] N. Akhtar and A. Mian, “Threat of adversarial attacks on deep learning in computer vision: A survey,” *IEEE Access*, vol. 6, pp. 14410–14430, 2018.
- [6] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, “A survey on 5g networks for the internet of things: Communication technologies and challenges,” *IEEE Access*, vol. 6, pp. 3619–3647, 2017.
- [7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [8] R. Al-Mutiri, M. Al-Rodhaan, and Y. Tian, “Improving vehicular authentication in vanet using cryptography,” *International Journal of Communication Networks and Information Security*, vol. 10, no. 1, pp. 248–255, 2018.
- [9] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, “Security vulnerabilities of connected vehicle streams and their impact on cooperative driving,” *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, 2015.
- [10] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis et al., “Understanding the mirai botnet,” in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 1093–1110.
- [11] J. Balasch, A. Rial, C. Troncoso, B. Preneel, I. Verbauwhede, and C. Geuens, “Pretp: Privacy-preserving electronic toll pricing,” in *USENIX Security Symposium*, vol. 10, 2010, pp. 63–78.
- [12] A. Basak, S. Bhunia, and S. Ray, “A Flexible Architecture for Systematic Implementation of SoC Security Policies,” in *ICCAD*, 2015, pp. 536–543.
- [13] —, “Exploiting design-for-debug for flexible SoC security architecture,” in *DAC*, 2016, pp. 167:1–167:6.
- [14] Y. O. Basciftci, F. Chen, J. Weston, R. Burton, and C. E. Koksall, “How vulnerable is vehicular communication to physical layer jamming attacks?” in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. IEEE, 2015, pp. 1–5.
- [15] A. Belmonte Martin, L. Marinos, E. Rekleitis, G. Spanoudakis, and N. Petroulakis, “Threat landscape and good practice guide for software defined networks/5g,” 2015.
- [16] S. Bhunia, S. Ray, and S. Sur-Kolay, *Fundamentals of IP and SoC Security: Design, Validation, and Debug*. Springer, 2017.
- [17] S. Bhunia and M. Tehranipoor, *The Hardware Trojan War: Attacks, Myths, and Defenses*. Springer, 2017.
- [18] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. T. Sukhvasi, C. Patel, and S. Geirhofer, “Network densification: the dominant theme for wireless evolution into 5g,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, 2014.
- [19] V. Bibhu, R. Kumar, B. S. Kumar, and D. K. Singh, “Performance analysis of black hole attack in vanet,” *International Journal Of Computer Network and Information Security*, vol. 4, no. 11, p. 47, 2012.
- [20] I. Bilogrevic, I. Aad, P. Ginzboorg, V. Niemi, J.-P. Hubaux et al., “Track me if you can: On the effectiveness of context-based identifier changes in deployed mobile networks,” in *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS 2012)*, no. CONF. Internet Society, 2012.
- [21] N. Bißmeyer, J. Njeukam, J. Petit, and K. M. Bayarou, “Central misbehavior evaluation for vanets based on mobility data plausibility,” in *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*. ACM, 2012, pp. 73–82.
- [22] S. Biswas, J. Mi, and V. Mi, “Ddos attack on wave-enabled vanet through synchronization,” in *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 1079–1084.
- [23] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, “Towards an open source ieee 802.11 p stack: A full sdr-based transceiver in gnu radio,” in *2013 IEEE Vehicular Networking Conference*. IEEE, 2013, pp. 143–149.
- [24] J. J. Blum and P. O. Okusun, “Privacy implications of the traffic probe message service,” in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 342–347.
- [25] S. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, and M. Szydlo, “Security analysis of a cryptographically-enabled rfid device,” in *USENIX Security Symposium*, vol. 31, 2005, pp. 1–16.
- [26] C. Miller, C. Valasek, “Securing Self-Driving Cars (one company at a time),” 2018, [Black Hat 2018, Briefing on Applied Self Driving Car Security].
- [27] J. A. L. Calvo and R. Mathar, “Secure blockchain-based communication scheme for connected vehicles,” in *2018 European Conference on Networks and Communications (EuCNC)*. IEEE, 2018, pp. 347–351.
- [28] C. Cerrudo and D. Spaniel, “Keeping smart cities smart: preempting emerging cyber attacks in us cities,” *Institute for Critical Infrastructure Technology*, 2015.
- [29] A. C.-F. Chan and J. Zhou, “On smart grid cybersecurity standardization: Issues of designing with nistir 7628,” *IEEE Communications Magazine*, vol. 51, no. 1, pp. 58–65, 2013.
- [30] —, “Cyber–physical device authentication for the smart grid electric vehicle ecosystem,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 7, pp. 1509–1517, 2014.
- [31] 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.
- [32] C. Chen, X. Wang, W. Han, and B. Zang, “A robust detection of the sybil attack in urban vanets,” in *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*. IEEE, 2009, pp. 270–276.
- [33] Q. A. Chen, Y. Yin, Y. Feng, Z. M. Mao, and H. X. Liu, “Exposing congestion attack on emerging connected vehicle based traffic signal control,” in *Network and Distributed Systems Security (NDSS) Symposium*, 2018.

- [34] W. Chen, S. Ray, J. Bhadra, M. S. Abadir, and L. Wang, "Challenges and Trends in Modern SoC Design Verification," *IEEE Design & Test*, vol. 34, no. 5, pp. 7–22, 2017.
- [35] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *USENIX Security Symposium*, 2016.
- [36] J. Choi and S. Jung, "A security framework with strong non-repudiation and privacy in vanets," in *2009 6th IEEE Consumer Communications and Networking Conference*. IEEE, 2009, pp. 1–5.
- [37] G. Clark, M. Doran, and W. Glisson, "A malicious attack on the machine learning policy of a robotic system," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 516–521.
- [38] M. Conner, "Friend or foe: Battery-authentication ics separate the good guys from the bad-all battery packs are not created equal: Unauthorized after-market packs may contain cells that can self-destruct when," *EDN*, vol. 51, no. 3, pp. 59–64, 2006.
- [39] S. Daneshmand, A. Jafarnia-Jahromi, A. Broumandan, and G. Lachapelle, "A low-complexity gps anti-spoofing method using a multi-antenna array," *aa*, vol. 2, p. 2, 2012.
- [40] D. Davidson, H. Wu, R. Jelinek, V. Singh, and T. Ristenpart, "Controlling uavs with sensor input spoofing attacks," in *10th {USENIX} Workshop on Offensive Technologies ({WOOT} 16)*, 2016.
- [41] S. L. Dennisen and J. P. Müller, "Agent-based voting architecture for traffic applications," in *German Conference on Multiagent System Technologies*. Springer, 2015, pp. 200–217.
- [42] T. Dobbyn, "U.S. opening formal probe into GM Volt fire risk," Sep 2011. [Online]. Available: <https://www.reuters.com/article/us-gm-volt/u-s-opening-formal-probe-into-gm-volt-fire-risk-idUSTR7A01SH20111126>
- [43] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.
- [44] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Hummingbird: ultra-lightweight cryptography for resource-constrained devices," in *International Conference on Financial Cryptography and Data Security*. Springer, 2010, pp. 3–18.
- [45] I. ETSI, "Intelligent transport systems (its); security; threat, vulnerability and risk analysis (tvr);" ETSI TR 102 893, European Telecommunications Standards Institute, Tech. Rep., 2010.
- [46] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Models," *arXiv*, Jul 2017. [Online]. Available: <https://arxiv.org/abs/1707.08945>
- [47] M. Feiri, J. Petit, and F. Kargl, "Efficient and secure storage of private keys for pseudonymous vehicular communication," in *Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles*. ACM, 2013, pp. 9–18.
- [48] A. Filippi, K. Moerman, V. Martinez, A. Turley, O. Haran, and R. Toledano, "Ieee 802.11 p ahead of lte-v2v for safety applications," *Autotalks NXP*, 2017.
- [49] U. Fiore, F. Palmieri, A. Castiglione, V. Loia, and A. De Santis, "Multimedia-based battery drain attacks for android devices," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2014, pp. 145–150.
- [50] I. D. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and vulnerable: A story of telematic failures," in *USENIX Workshop on Offensive Technologies*, 2015.
- [51] 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)*. Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2011.
- [52] M. T. Garip, M. E. Gursoy, P. Reiher, and M. Gerla, "Congestion attacks to autonomous cars using vehicular botnets," in *NDSS Workshop on Security of Emerging Networking Technologies (SENT), San Diego, CA*, 2015.
- [53] M. Ghaderi, D. Goeckel, A. Orda, and M. Dehghan, "Efficient wireless security through jamming, coding and routing," in *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*. IEEE, 2013, pp. 505–513.
- [54] B. Ghena, W. Beyer, A. Hillaker, J. Pevanek, and J. A. Halderman, "Green lights forever: Analyzing the security of traffic infrastructure," in *8th {USENIX} Workshop on Offensive Technologies ({WOOT} 14)*, 2014.
- [55] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in vanets," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. ACM, 2004, pp. 29–37.
- [56] A. Greenberg, "After jeep hack, chrysler recalls 1.4m vehicles for bug fix," 2015. [Online]. Available: <https://www.wired.com/2015/07/jeep-hack-chrysler-recalls-1-4m-vehicles-bug-fix/>
- [57] S. J. Greenwald, "Discussion Topic: What is the Old Security Paradigm," in *Workshop on New Security Paradigms*, 1998, pp. 107–118.
- [58] I. A. Grzembera, *MOST: the automotive multimedia network*. Franzis Verlag, 2012.
- [59] G. Guette and C. Bryce, "Using tpms to secure vehicular ad-hoc networks (vanets)," in *IFIP International Workshop on Information Security Theory and Practices*. Springer, 2008, pp. 106–116.
- [60] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: detection, avoidance, and the challenges ahead," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 9–23, 2014.
- [61] Y. Hao, J. Tang, and Y. Cheng, "Cooperative sybil attack detection for position based applications in privacy preserved vanets," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE, 2011, pp. 1–5.
- [62] S. K. Harit, G. Singh, and N. Tyagi, "Fox-hole model for data-centric misbehaviour detection in vanets," in *2012 Third International Conference on Computer and Communication Technology*. IEEE, 2012, pp. 271–277.
- [63] M. Harris, "Researcher hacks self-driving car sensors," *IEEE Spectrum*, vol. 9, 2015.
- [64] C. Harsch, A. Festag, and P. Papadimitratos, "Secure position-based routing for vanets," in *2007 IEEE 66th Vehicular Technology Conference*. IEEE, 2007, pp. 26–30.
- [65] H. Hasbullah, I. A. Soomro *et al.*, "Denial of service (dos) attack and its possible solutions in vanet," *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 4, no. 5, pp. 813–817, 2010.
- [66] M. Heller, "Pegasus ios exploit uses three zero days to attack high-value targets," <https://searchsecurity.techtarget.com/news/450303267/Pegasus-iOS-exploit-uses-three-zero-days-to-attack-high-value-targets>, (Accessed on 09/19/2019).
- [67] J. Hortelano, J. C. Ruiz, and P. Manzoni, "Evaluating the usefulness of watchdogs for intrusion detection in vanets," in *2010 IEEE International Conference on Communications Workshops*. IEEE, 2010, pp. 1–5.
- [68] L. Huang and Q. Yang, "Low-cost gps simulator gps spoofing by sdr," 2015.
- [69] T. Huddleston Jr. These chinese hackers tricked tesla's autopilot into suddenly switching lanes. CNBC. [Online]. Available: <https://www.cnbc.com/2019/04/03/chinese-hackers-tricked-teslas-autopilot-into-switching-lanes.html>
- [70] T. E. Humphreys, "Detection strategy for cryptographic gnss anti-spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1073–1090, 2013.
- [71] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the spoofing threat: Development of a portable gps civilian spoofer," in *Radionavigation laboratory conference proceedings*, 2008.
- [72] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues and future challenges," *IEEE Communications Surveys & Tutorials*, 2018.
- [73] O. A. Ibrahim, A. M. Hussain, G. Oligeri, and R. Di Pietro, "Key is in the air: Hacking remote keyless entry systems," 2018.
- [74] J. T. Isaac, J. S. Camara, S. Zeadally, and J. T. Marquez, "A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks," *Computer Communications*, vol. 31, no. 10, pp. 2478–2484, 2008.
- [75] R. M. Ishtiaq Roufa, H. Mustafaa, S. O. Travis Taylor, W. Xua, M. Gruteserb, W. Trappeb, and I. Seskarb, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *19th USENIX Security Symposium, Washington DC*, 2010, pp. 11–13.
- [76] T. Jeske, "Floating car data from smartphones: What google and waze know about you and how hackers can control traffic," *Proc. of the BlackHat Europe*, pp. 1–12, 2013.
- [77] A. Jindal, A. Pathak, Y. C. Hu, and S. Midkiff, "On death, taxes, and sleep disorder bugs in smartphones," in *Proceedings of the Workshop on Power-Aware Computing and Systems*. ACM, 2013, p. 1.

- [78] Kali Tools, "Binwalk Penetration Testing Tools." [Online]. Available: <https://tools.kali.org/forensics/binwalk>
- [79] S. Kamkar, "Drive it like you hacked it: New attacks and tools to wirelessly steal cars," *Presentation at DEFCON*, vol. 23, 2015.
- [80] H. Kaur, S. Batish, and A. Kakaria, "An approach to detect the wormhole attack in vehicular adhoc networks," *Int. J. Smart Sens. Ad Hoc Netw.*, vol. 4, pp. 86–89, 2012.
- [81] L. Kelion, "Nissan Leaf electric car hack revealed," Sep 2019. [Online]. Available: <https://www.bbc.com/news/technology-35642749>
- [82] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.
- [83] R. Khatoun, P. Gut, R. Doulami, L. Khoukhi, and A. Serhrouchni, "A reputation system for detection of black hole attack in vehicular networking," in *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE, 2015, pp. 1–5.
- [84] G.-H. Kim, K.-H. Lee, S.-S. Kim, and J.-M. Kim, "Vehicle relay attack avoidance methods using rf signal strength," *Communications and Network*, vol. 5, no. 03, p. 573, 2013.
- [85] J. Klein, "Parked Teslas Keep Catching on Fire Randomly, And There's No Recall In Sight," Sep 2019. [Online]. Available: <https://www.thedrive.com/news/28420/parked-teslas-keep-catching-on-fire-randomly-and-theres-no-recall-in-sight>
- [86] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.
- [87] 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.
- [88] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv*, Jul 2016. [Online]. Available: <https://arxiv.org/abs/1607.02533>
- [89] D. Kushwaha, P. K. Shukla, and R. Baraskar, "A survey on sybil attack in vehicular ad-hoc network," *International Journal of Computer Applications*, vol. 98, no. 15, 2014.
- [90] L. Johnson, M. Fitzsimmons, "Uber self-driving cars: Everything you need to know," <https://www.techradar.com/news/uber-self-driving-cars>, 2018, [Online article on Techradar].
- [91] F. Lambert, "Tesla Model S caught fire and burned down while charging at a Supercharger," Jan 2016. [Online]. Available: <https://electrek.co/2016/01/01/tesla-model-s-caught-fire-and-burned-down-charging-supercharger>
- [92] —, "Tesla vehicle caught on fire while plugged in at Supercharger station - Electrek," Jun 2019. [Online]. Available: <https://electrek.co/2019/06/01/tesla-fire-supercharger>
- [93] N. Lawson, "Highway to hell: Hacking toll systems," *Presentation at Blackhat*, 2008.
- [94] R. Leale and K. Sireci Renner. (2019) Car hacking village: Securing critical automotive systems. [Online]. Available: <https://www.carhackingvillage.com/>
- [95] P. LeBeau, "Waymo hits 10 millionth mile, prepares for public ride hailing," <https://www.cnbc.com/2018/10/10/waymo-hits-10-millionth-mile-prepares-for-public-ride-hailing.html>, 2018, [Online article on CNBC].
- [96] K. C. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla, "First experience with cartorrent in a real vehicular ad hoc network testbed," in *2007 Mobile Networking for Vehicular Environments*. IEEE, 2007, pp. 109–114.
- [97] A. Lima, F. Rocha, M. Völöp, and P. Esteves-Veríssimo, "Towards safe and secure autonomous and cooperative vehicle ecosystems," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*. ACM, 2016, pp. 59–70.
- [98] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen, "Security in vehicular ad hoc networks," *IEEE communications magazine*, vol. 46, no. 4, pp. 88–95, 2008.
- [99] X. Lin, P. Bogdan, N. Chang, and M. Pedram, "Machine learning-based energy management in a hybrid electric vehicle to minimize total operating cost," *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 627–634, Nov 2015.
- [100] C. B. Liu, B. Sadeghi, and E. W. Knightly, "Enabling vehicular visible light communication (v2lc) networks," in *Proceedings of the Eighth ACM international workshop on Vehicular inter-networking*. ACM, 2011, pp. 41–50.
- [101] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, *A Comprehensive Guide to 5G Security*. John Wiley & Sons, 2018.
- [102] A. B. Lopez, K. Vatanparvar, A. P. D. Nath, S. Yang, S. Bhunia, and M. A. Al Faruque, "A security perspective on battery systems of the internet of things," *Journal of Hardware and Systems Security*, vol. 1, no. 2, pp. 188–199, 2017.
- [103] S. Lu, Y. Yao, and W. Shi, "Collaborative learning on the edges: A case study on connected vehicles," in *2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19)*, 2019.
- [104] S. Lucero, "ihs-technology-whitepaper-cellular-vehicle-to-everything-c-v2x-connectivity.pdf," <https://www.qualcomm.com/media/documents/files/ihs-technology-whitepaper-cellular-vehicle-to-everything-c-v2x-connectivity.pdf>, June 2016, (Accessed on 09/19/2019).
- [105] S. Lukic and Z. Pantic, "Cutting the cord: Static and dynamic inductive charging of electric vehicles," *IEEE Electrification Magazine*, vol. 1, no. 1, pp. 57–64, 2013.
- [106] R. Makowitz and C. Temple, "Flexray-a communication network for automotive control systems," in *2006 IEEE International Workshop on Factory Communication Systems*. IEEE, 2006, pp. 207–212.
- [107] T. R. Markham and A. Chernoguzov, "A balanced approach for securing the obd-ii port," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 10, no. 2017-01-1662, pp. 390–399, 2017.
- [108] S. Meiklejohn, K. Mowery, S. Checkoway, and H. Shacham, "The phantom tollbooth: Privacy-preserving electronic toll collection in the presence of driver collusion," in *USENIX security symposium*, vol. 201, no. 1, 2011.
- [109] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *Black Hat USA*, vol. 2014, p. 94, 2014.
- [110] —. (2015) Car hacking: the definitive source. [Online]. Available: <http://illmatics.com/content.zip>
- [111] —, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, p. 91, 2015.
- [112] P. Mishra, S. Bhunia, and M. Tehranipoor, *Hardware IP Security and Trust*. Springer, 2017.
- [113] R. N. Mitra and D. P. Agrawal, "5g mobile technology: A survey," *ICT Express*, vol. 1, no. 3, pp. 132–137, 2015.
- [114] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, D. Boneh *et al.*, "Location privacy via private proximity testing," in *NDSS*, vol. 11, 2011.
- [115] T. P. Oman and K. J. Hawes, "Relay attack prevention for passive entry passive start (peps) vehicle security systems," Jan. 6 2015, uS Patent 8,930,045.
- [116] H. Onishi, "Paradigm change of vehicle cyber security," in *2012 4th International Conference on Cyber Conflict (CYCON 2012)*. IEEE, 2012, pp. 1–11.
- [117] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka *et al.*, "Scenarios for 5g mobile and wireless communications: the vision of the metis project," *IEEE communications magazine*, vol. 52, no. 5, pp. 26–35, 2014.
- [118] S. Park, B. Aslam, D. Turgut, and C. C. Zou, "Defense against sybil attack in vehicular ad hoc network based on roadside unit support," in *MILCOM 2009-2009 IEEE Military Communications Conference*. IEEE, 2009, pp. 1–7.
- [119] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2015.
- [120] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.
- [121] S. Plosz and P. Varga, "Security and safety risk analysis of vision guided autonomous vehicles," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2018, pp. 193–198.
- [122] Y. Qian, K. Lu, and N. Moayeri, "A secure vanet mac protocol for dsrc applications," in *IEEE GLOBECOM 2008-2008 IEEE Global Communications Conference*. IEEE, 2008, pp. 1–5.
- [123] B. Qin, Q. Wu, J. Domingo-Ferrer, and L. Zhang, "Preserving security and privacy in large-scale vanets," in *International Conference on Information and Communications Security*. Springer, 2011, pp. 121–135.
- [124] A. Rawat, S. Sharma, and R. Sushil, "Vanet: Security attacks and its possible solutions," *Journal of Information and Operations Management*, vol. 3, no. 1, p. 301, 2012.
- [125] S. Ray, "Transportation security in the era of autonomous vehicles: Challenges and practice," in *ICCAD 2017*, 2017.
- [126] —, "Safety, Security, and Reliability: The Automotive Robustness Problem and an Architectural Solution," in *ICCE 2019*, 2019.

- [127] S. Ray, W. Chen, J. Bhadra, and M. A. A. Faruque, "Extensibility in Automotive Security: Current Practice and Challenges," in *DAC 2017*, 2017.
- [128] S. Ray, W. Chen, and R. Cammarota, "Protecting the Supply Chain of Automotives and IoTs," in *DAC 2018*, 2017, pp. 89:1–89:4.
- [129] S. Ray, E. Peeters, M. Tehranipoor, and S. Bhunia, "System-on-Chip Platform Security Assurance: Architecture and Validation," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 21–37, 2018.
- [130] J. Reilly, S. Martin, M. Payer, and A. Bayen, "On cybersecurity of freeway control systems: Analysis of coordinated ramp metering attacks," *Transportation Research, Part B*, 2014.
- [131] J. Reilly, S. Martin, M. Payer, and A. M. Bayen, "Creating complex congestion patterns via multi-objective optimal freeway traffic control with application to cyber-security," *Transportation Research Part B: Methodological*, vol. 91, pp. 366–382, 2016.
- [132] C. Rohner, S. Raza, D. Puccinelli, and T. Voigt, "Security in visible light communication: Novel challenges and opportunities," *Sensors & Transducers Journal*, vol. 192, no. 9, pp. 9–15, 2015.
- [133] M. Ruff, "Evolution of local interconnect network (lin) solutions," in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484)*, vol. 5. IEEE, 2003, pp. 3382–3389.
- [134] A. Sargolzaei, C. D. Crane, A. Abbaspour, and S. Noei, "A machine learning approach for fault detection in vehicular cyber-physical systems," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 636–640.
- [135] T. Schütze, "Automotive security: Cryptography for car2x communication," in *Embedded World Conference*. Citeseer, 2011.
- [136] K. Seiberts and J. Childers, "Relay attack prevention for passive entry/passive start systems," Aug. 11 2015, uS Patent 9,102,296.
- [137] R. Sens, "Be ready to fight new 5g vulnerabilities," *Network Security*, vol. 2018, no. 10, pp. 6–7, 2018.
- [138] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2013, pp. 55–72.
- [139] Y. Shoukry, S. Mishra, Z. Luo, and S. Diggavi, "Sybil attack resilient traffic networks: A physics-based trust propagation approach," in *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*. IEEE Press, 2018, pp. 43–54.
- [140] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, "Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4917–4932, 2017.
- [141] R. Shrestha, R. Bajracharya, and S. Y. Nam, "Blockchain-based message dissemination in vanet," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. IEEE, 2018, pp. 161–166.
- [142] Y. L. Sit, C. Sturm, L. Reichardt, T. Zwick, and W. Wiesbeck, "The ofdm joint radar-communication system: An overview," in *Proc. Int. Conf. Advances in Satellite and Space Communications (SPACOMM 2011)*, 2011, pp. 69–74.
- [143] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "Darts: Deceiving autonomous cars with toxic signs," *arXiv preprint arXiv:1802.06430*, 2018.
- [144] C. Smith, *The car hacker's handbook: a guide for the penetration tester*. no starch Press, 2016.
- [145] C. Specification, "v2. 0," *Common public radio interface (CPRI)*, pp. 1–75, 2004.
- [146] I. A. Sumra, J.-L. Ab Manan, and H. Hasbullah, "Timing attack in vehicular network," in *Proceedings of the 15th WSEAS International Conference on Computers, World Scientific and Engineering Academy and Society (WSEAS), Corfu Island, Greece*, 2011, pp. 151–155.
- [147] J. Sun, C. Zhang, and Y. Fang, "An id-based framework achieving privacy and non-repudiation in vehicular ad hoc networks," in *MILCOM 2007-IEEE Military Communications Conference*. IEEE, 2007, pp. 1–7.
- [148] A. Takanen, J. D. DeMott, and C. Mille, *Fuzzing for Software Security Testing and Quality Assurance*. Artech House, 2008.
- [149] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "Ibe-lite: a lightweight identity-based cryptography for body sensor networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 926–932, 2009.
- [150] T. Tanizawa, T. Suzumiya, and K. Ikeda, "Cloud-connected battery management system supporting e-mobility," *Fujitsu Sci. Tech. J.*, vol. 51, no. 4, pp. 27–35, 2015.
- [151] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive can bus," in *WCICSS*, 2015, pp. 45–49.
- [152] A. N. S. Team, "A detailed analysis of the malware responsible for global iot botnets and massive ddos attacks," <https://www.a10networks.com/marketing-comms/white-papers/investigating-mirai/>, (Accessed on 09/19/2019).
- [153] M. Tehranipoor and C. Wang, *Introduction to Hardware Security and Trust*. Springer, 2011.
- [154] (2019) Experimental security research of tesla autopilot. Tencent Security. [Online]. Available: https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf
- [155] (2016) Car hacking research: Remote attack tesla motors. Tencent Security: Keen Security Lab. [Online]. Available: <https://keenlab.tencent.com/en/2016/09/19/Keen-Security-Lab-of-Tencent-Car-Hacking-Research-Remote-Attack-to-Tesla-Cars>
- [156] (2017) New car hacking research: 2017, remote attack tesla motors again. Tencent Security: Keen Security Lab. [Online]. Available: <https://keenlab.tencent.com/en/2017/07/27/New-Car-Hacking-Research-2017-Remote-Attack-Tesla-Motors-Again>
- [157] (2018) New vehicle security research by keenlab: Experimental security assessment of bmw cars. Tencent Security: Keen Security Lab. [Online]. Available: <https://keenlab.tencent.com/en/2018/05/22/New-CarHacking-Research-by-KeenLab-Experimental-Security-Assessment-of-BMW-Cars>
- [158] (2019) Keen security lab blog. Tencent Security: Keen Security Lab. [Online]. Available: <https://keenlab.tencent.com/en/>
- [159] I. Ullah and S. U. Rehman, "Analysis of black hole attack on manets using different manet routing protocols," 2010.
- [160] M. Uysal, Z. Ghassemloooy, A. Bekkali, A. Kadri, and H. Menour, "Visible light communication for vehicular networking: performance study of a v2v system using a measured headlamp beam pattern model," *IEEE Vehicular Technology Magazine*, vol. 10, no. 4, pp. 45–53, 2015.
- [161] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth-a simple, backward compatible broadcast authentication protocol for can bus," in *ECRYPT Workshop on Lightweight Cryptography*, vol. 2011, 2011.
- [162] M. Vasek, M. Thornton, and T. Moore, "Empirical analysis of denial-of-service attacks in the bitcoin ecosystem," in *International conference on financial cryptography and data security*. Springer, 2014, pp. 57–71.
- [163] S. Verma, B. Mallick, and P. Verma, "Impact of gray hole attack in vanet," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*. IEEE, 2015, pp. 127–130.
- [164] J. Wan, A. Lopez, and M. A. A. Faruque, "Physical layer key generation: Securing wireless communication in automotive cyber-physical systems," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 2, p. 13, 2018.
- [165] J. Wan, A. B. Lopez, and M. A. Al Faruque, "Exploiting wireless channel randomness to generate keys for automotive cyber-physical system security," in *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICPPS)*. IEEE, 2016, pp. 1–10.
- [166] L. Whitney, "Viking horde malware attacks android devices - cnet," <https://www.cnet.com/news/viking-horde-malware-attacks-android-devices/>, (Accessed on 09/19/2019).
- [167] M. Wolf, "Report of the nsf workshop on internet-of-things (iot) systems," Tech. Rep., November 2019.
- [168] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*, 2004.
- [169] A. M. Wyglinski, X. Huang, T. Padir, L. Lai, T. R. Eisenbarth, and K. Venkatasubramanian, "Security of autonomous systems employing embedded computing and sensors," *IEEE micro*, vol. 33, no. 1, pp. 80–86, 2013.
- [170] X. Xiong, D. S. Wong, and X. Deng, "Tinypairing: A fast and lightweight pairing-based cryptographic library for wireless sensor networks," in *2010 IEEE Wireless Communication and Networking Conference*. IEEE, 2010, pp. 1–6.
- [171] C. Xu, H. Liu, P. Li, and P. Wang, "A remote attestation security model based on privacy-preserving blockchain for v2x," *IEEE Access*, vol. 6, pp. 67 809–67 818, 2018.
- [172] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, 2016.
- [173] N. Yang, L. Wang, G. Geraci, M. El-kashlan, J. Yuan, and M. Di Renzo, "Safeguarding 5g wireless communication networks using physical layer security," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 20–27, 2015.

- [174] T. Yang, L. Kong, W. Xin, J. Hu, and Z. Chen, "Resisting relay attacks on vehicular passive keyless entry and start systems," in *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, 2012, pp. 2232–2236.
- [175] C. Yuan, J. Thai, and A. M. Bayen, "Zubers against zlyfts apocalypse: An analysis framework for dos attacks on mobility-as-a-service systems," in *Proceedings of the 7th International Conference on Cyber-Physical Systems*. IEEE Press, 2016, p. 24.
- [176] K. Zaidi, M. Milojevic, V. Rakocevic, and M. Rajarajan, "Data-centric rogue node detection in vanets," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2014, pp. 398–405.
- [177] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All your {GPS} are belong to us: Towards stealthy manipulation of road navigation systems," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1527–1544.
- [178] T. Zhang, H. Antunes, and S. Aggarwal, "Defending connected vehicles against malware: Challenges and a solution framework," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 10–21, 2014.
- [179] T. Zhou, R. R. Choudhury, P. Ning, and K. Chakrabarty, "P2dapsybil attacks detection in vehicular ad hoc networks," *IEEE journal on selected areas in communications*, vol. 29, no. 3, pp. 582–594, 2011.
- [180] T. Ziermann, S. Wildermann, and J. Teich, "Can+: A new backward-compatible controller area network (can) protocol with up to 16x higher data rates," in *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2009, pp. 1088–1093.