Autonomous Driving Security: State of the Art and Challenges

Cong Gao, Geng Wang, Weisong Shi, *Fellow, IEEE*, Zhongmin Wang, and Yanping Chen

***Abstract*—The autonomous driving industry has mushroomed over the past decade. Although autonomous driving has undoubtedly become one of the most promising technologies of this century, its development faces multiple challenges, of which security is the major concern. In this article, we present a thorough analysis of autonomous driving security. First, the attack surface of autonomous driving is presented. After an analysis of the operation of autonomous driving in terms of key components and technologies, the security of autonomous driving is elaborated in four dimensions: 1) sensors; 2) operating system; 3) control system; and 4) vehicle-to-everything (V2X) communication. Sensor security is examined from five components, which are mainly responsible for self-positioning and environmental perception. The analysis of operating system security, the second dimension, is concentrated on the robot operating system. Concerning the control system security, the controller area network is approached mainly from vulnerabilities and protection measures. The fourth dimension, V2X communication security, is probed from four categories of attacks: 1) authenticity/identification; 2) availability; 3) data integrity; and 4) confidentiality with corresponding solutions. Moreover, the drawbacks of existing methods adopted in the four dimensions are also provided. Finally, a conceptual multilayer defense framework is proposed to secure the information flow from external communication to the physical autonomous vehicle.**

***Index Terms*—Attack surface, autonomous driving, control area network, data distribution service (DDS), robot operating system, security, sensor, unmanned vehicle, vehicle-to-everything (V2X) communication.**

I. Introduction

With the rapid improvement of intelligent vehicles, autonomous driving has attracted much research attention. Autonomous vehicles are considered to be beneficial for alleviating traffic congestion and reducing the number of road accidents. However, current autonomous driving technologies are immature and still in development. The safety of the passengers and the vehicle itself is far from guaranteed [1], [2]. For instance, in 2018, an Uber unmanned vehicle collided with a pedestrian wheeling a bicycle across the road during a road test in Arizona [3]. This was the world's first case of an autonomous vehicle accident, which caused the death of a pedestrian. The incident subsequently led to a stormy discussion of the safety of autonomous vehicles.

Manuscript received July 15, 2021; revised October 4, 2021; accepted November 19, 2021. Date of publication November 23, 2021; date of current version May 9, 2022. This work was supported in part by the Science and Technology Project of the Shaanxi Provincial Science and Technology Department, China, under Grant 2019ZDLGY07-08; in part by the Scientific Research Program Funded by Shaanxi Provincial Education Department, China, under Grant 21JP115; and in part by the Special Funds for Construction of Key Disciplines in Universities in Shaanxi, China. *(Corresponding author: Cong Gao.)*

Cong Gao and Geng Wang are with the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (e-mail: cgao@xupt.edu.cn; gwang_xupt@126.com).

Weisong Shi is with the College of Engineering, Wayne State University, Detroit, MI 48202 USA (e-mail: weisong@wayne.edu).

Zhongmin Wang and Yanping Chen are with the Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (e-mail: zmwang@xupt.edu.cn; chenyp@xupt.edu.cn).

Digital Object Identifier 10.1109/JIOT.2021.3130054

A. Autonomous Driving Security

An autonomous vehicle is a comprehensive system, which mainly consists of a positioning system, a perception system, a planning system, and a control system [4]. The security of autonomous vehicles generally refers to the security during the driving process, including the security of the sensor, operating system, control system, and vehicle-to-everything (V2X) communication.

1) Sensor Security: Sensor security mainly deals with the security of the actual components, such as the onboard sensors and onboard chips. For instance, Google's self-driving vehicles employ a variety of sensors to detect the driving environment. The collected sensor data are used to analyze whether a vehicle is in a safe driving state.

2) Operating System Security: Operating system security refers to ensuring the integrity and availability of the operating system and preventing unauthorized access. At present, most autonomous vehicles are developed based on a robot system. For instance, Baidu's autonomous vehicle platform Apollo [5] is based on the most famous robot operating system, ROS [6]. ROS is a robot middleware platform that provides the basic functions of an operating system for heterogeneous computer clusters. However, ROS was originally designed without considering security. Other similar operating systems also suffer from this problem.

3) Control System Security: Control system security guarantees that the onboard decision-making system gives correct instructions for steering, acceleration, deceleration, and parking of the autonomous vehicle based on the data collected from both the environment and the vehicle itself. However, with the increasing variety of external interfaces of a vehicle, novel attack surfaces keep emerging. Thus, the control system is vulnerable to illegal invasions.

4) V2X Communication Security: V2X communication security refers to the security of the communication of vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I),

vehicle-to-pedestrian (V2P), and vehicle-to-network (V2N). The design of a vehicle network system is supposed to guarantee the above communication against attacks. Moreover, the information about surrounding vehicles and environmental conditions coming from V2X communication further contributes to the security of a vehicle.

B. Attack Surface

The notion of *attack surface* usually attributed to Michael Howard of Microsoft. It is informally introduced to act as an indicator of the security of a software system [7].

Early research on attack surface [8]–[12] mainly focused on software systems and laid a solid foundation for subsequent study. Michael Howard considered that attack surface is a set of attack features: open sockets, open RPC endpoints, open named pipes, services, etc., [7]. Manadhata *et al.* [12] presented the definition that a system's attack surface is the subset of resources that an attacker can use to attack the system.

Ren *et al.* [2] briefly categorized security threats surrounding an autonomous vehicle into three groups of attacks surfaces: 1) various sensors; 2) in-vehicle access and control systems; and 3) in-vehicle network protocols.

Recent literature about attack surface focused on creating empirical and theoretical measures for the attack surface of a software system or computer network [13], such as [14]–[17].

In the field of autonomous driving, notable literatures concerning attack surface are as follows.

Maple *et al.* [18] developed a reference architecture using a hybrid functional-communication viewpoint for attack surface analysis of connected autonomous vehicles (CAVs).

Salfer and Eckert [19] proposed a method for the attack surface and vulnerability assessment automation of automotive electronic control units (ECUs) based on development data and software flash images.

Checkoway *et al.* [20] conducted a detailed analysis of the external attack surface for automobiles. This work mainly focused on remote compromise.

In [21], threat areas of in-vehicle infotainment systems were discussed. Seven vulnerabilities of Linux-based in-vehicle infotainment systems and 15 potential attack surfaces were identified.

Chattopadhyay *et al.* [22] developed a security-by-design framework for autonomous vehicles. The framework contains a high-level model, which defines the attack surfaces of autonomous vehicles into three layers.

Dominic *et al.* [23] presented a risk assessment framework for autonomous and cooperative automated driving. A threat model was proposed based on the threat model described by the national highway traffic safety administration (NHTSA) [24] and security requirements described by the E-safety vehicle intrusion protected applications (EVITA) project [25]. Attack surfaces were described in five categories: 1) inertial/odometric; 2) range sensors; 3) global positioning system (GPS); 4) map update; and 5) V2V/V2I.

Petit and Shladover [26] studied the potential cyber attacks against automated vehicles. The attack surfaces in autonomous automated vehicles and cooperative automated vehicles were analyzed, respectively.

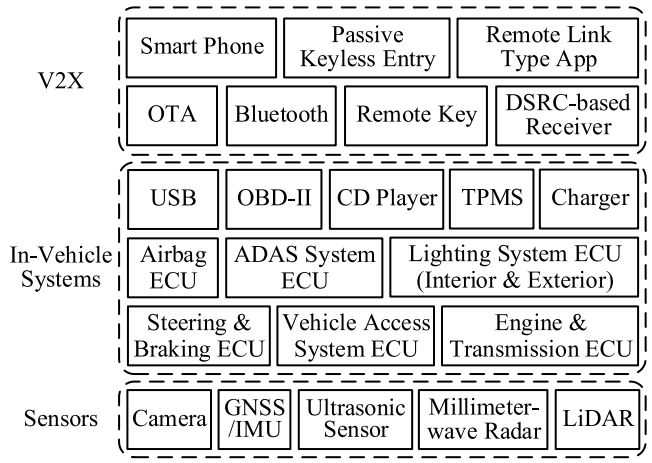

Fig. 1. Attack surfaces of autonomous driving.

Based on the analysis of the above literatures, we broadly divide the attack surfaces of autonomous driving into three categories. As shown in Fig. 1, they are sensors, in-vehicle systems, and V2X. For sensors: GNSS/IMU stands for global navigation satellite system and inertial measurement unit. LiDAR is short for light detection and ranging. For in-vehicle systems: OBD-II is short for the second generation of onboard diagnostics. TPMS stands for tire pressure monitoring system. ADAS is short for the advanced driving assistance system. For V2X: OTA stands for over-the-air. It is essentially just a synonym for wireless. DSRC is short for dedicated short-range communication. Fig. 1 is by no means exhaustive but aims to raise the security issues of autonomous vehicles.

C. Content and Roadmap

In this article, we review the state of the art and challenges involving the above four aspects of autonomous driving and point out the drawbacks of existing solutions. The main components and related technologies of autonomous driving are presented. The discussion of sensor security is focused on the cameras, GNSS/IMUs, ultrasonic sensors, millimeter-wave radar, and LiDAR. The discussion of operating system security is focused on ROS. A security enhancement data distribution service (DDS) adopted by ROS version 2 is described in detail. The analysis of control system security is focused on the controller area network (CAN). The vulnerabilities of CAN are analyzed based on five attack paths: 1) OBD-II; 2) electronic vehicle charger; 3) CD player; 4) TPMS; and 5) Bluetooth. Two types of protection methods are presented: 1) those based on encryption/authentication and 2) those based on intrusion detection. The recent development of the control area network standard is also presented based on CAN with flexible data rate (CAN FD). V2X communication security is analyzed based on four categories of attacks: 1) authenticity/identification; 2) availability; 3) data integrity; and 4) confidentiality. Moreover, the blockchain-based security measures for vehicular network are reviewed.

Finally, six real-world security incidents of autonomous vehicles are presented. Then, a conceptual multilayer defense framework for the security of autonomous driving is proposed.

The remainder of this article is structured as follows. In Section II, we review the main components and technologies of an autonomous driving system. In Section III, we discuss the security of five key sensors for autonomous vehicles. In Section IV, we analyze the security of the popular operating systems for autonomous vehicles. The discussion is concentrated on ROS, which plays a dominant role in the field of autonomous driving. In Section V, we discuss the security of control systems based on CAN. Vulnerabilities, attacks, and protections of CAN are presented. New standard of CAN is presented based on CAN FD. In Section VI, we summarize attacks against the communication in the Internet of Vehicles (IoVs) and the corresponding solutions. In Section VII-A, six real-world security incidents of autonomous vehicles are introduced. These incidents are presented in four categories: 1) sensor security; 2) operating system security; 3) control system security; and 4) V2X communication security. In Section VII-B, we propose a conceptual defense framework for automotive information security. Finally, we present our conclusions in Section VIII.

II. Autonomous Driving Technologies

An autonomous driving system is a kind of intelligent system that realizes autonomous driving based on onboard computer systems. It is an integration of multiple technologies. Generally speaking, an autonomous driving system requires powerful computing ability. The computing resources are responsible for the realization of the vehicle positioning, environmental perception, path planning, motion control, etc. For instance, Xiao *et al.* [27] proposed a blockchain-based algorithm called DAER to allocate resources for intensive computing tasks. In general, the realization of an autonomous driving system is based on multisensor information fusion and should meet the requirements of high performance and high security. The security of the related technologies for autonomous driving is a prerequisite for ensuring the security of autonomous vehicles on the road.

The autonomous driving technology stack is shown in Fig. 2. There are two major aspects: 1) components and 2) technologies.

A. Components

The key components of autonomous driving include GNSS/IMUs, sensors, V2X, and actuators. The GNSS/IMU is critical in the localization. It is a core component for sensor fusion and safe driving. Sensors play a pivotal role in environmental perception. Therefore, sensors should be deployed around an autonomous vehicle. The detection coverage of cooperative homogeneous sensors is often made to be overlapping so as to provide redundancy and accuracy. Different sensors use different detection technologies to perceive specific environmental information. An environment model is

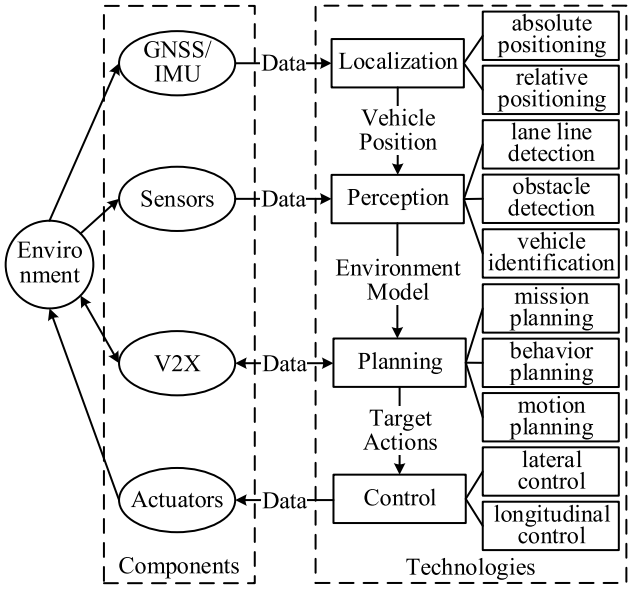

Fig. 2. Technology stack of autonomous driving.

built based on this information. For instance, V2X is able to collect real-time information about the surrounding vehicles and environmental conditions. This information is used for planning, which is critical in reducing traffic jams and enhancing the safety of the driving. Target actions given by the planning process are based on the information related to V2X and the model of the environment. The control module issues commands, in accordance with the actions aimed at, to the corresponding actuators. An actuator acts on the environment and changes the status of the vehicle. The technologies mainly involve localization, perception, planning, and control.

B. Technologies

1) Localization: Existing solutions to the localization of autonomous vehicles fall into two groups: 1) a vehicle networking solution based on V2X with shared location information and 2) a single agent solution based on multisensor information fusion. To ensure the safe and reliable operation of autonomous vehicles on the road, the accurate positioning of the vehicles is a prerequisite.

As one of the core functions of vehicle sensing systems, positioning plays an extremely important role in research into autonomous vehicles. In other words, positioning is a fundamental problem in this research area. The GNSS/IMU package is an effective solution for positioning of autonomous vehicles [28]. However, this method is unable to achieve high-precision positioning when the GNSS signals are weak, such as in underground parking lots and urban areas surrounded with high-rise buildings. Besides, GNSS signals are easily interfered with by a GPS jammer [29]. Map-assisted positioning is another popular type of autonomous vehicle positioning method. Simultaneous localization and mapping (SLAM) [30] is an example of this kind of algorithm. This technology is also known as concurrent mapping and localization (CML). SLAM determines the current position of a

vehicle based on the observed environmental characteristics. However, during a long-distance movement, the deviation of the SLAM positioning gradually increases, thus resulting in an inaccurate positioning, which is unacceptable for certain application scenarios. The above problem with SLAM positioning is effectively addressed by employing LiDAR to construct a point cloud map of the area of interest in advance [31], [32]. Several *semantics* are added to the map, both automatically and manually, such as specific markings of the lane lines, the location of traffic lights, and traffic rules on different roads. This kind of semantic map is called a high definition (HD) map.

2) Perception: As the most challenging module in autonomous vehicles, the perception system directly affects the results given by planning system and control system. Conventional perception modules mainly utilize computer vision technologies to extract information of the driving environment. The obtained information is used to conduct lane lines detection, obstacle detection, vehicle recognition/tracking, etc.

Autonomous vehicles are equipped with a variety of sensors. Among these sensors, ultrasonic radar, millimeter-wave radar, LiDAR, and cameras can be considered as *vision* in a broad sense. Due to low response speed and low resolution, ultrasonic radars are typically used for coarse-grained occasions, such as car reversing aid alarm systems [33]. On the one hand, when a vehicle is running at a high speed, the performance of ultrasonic radar ranging is unable to catch up with the variation of displacement. On the other hand, as the scattering angle of an ultrasonic radar is large, the signal reflected back is weak especially for the measurement of a distant target. Hence, the decrease in measurement accuracy might be significant. Millimeter-wave radar and LiDAR are mainly responsible for the ranging of medium and long distances. LiDAR generally relies on multiple laser transmitters and receivers to build 3-D point cloud maps. These maps are used to achieve real-time environmental perception. Two distinct advantages of LiDAR ranging are high precision and long distance. However, the actual performance of LiDAR might be poor in certain weather conditions (e.g., rain, snow, and fog), since the straight laser is blocked by obstacles. A millimeter-wave radar emits radio waves to determine the position of a target. This kind of radar is hardly affected by harsh weather conditions; thus, it is better than LiDAR in this respect. However, millimeter-wave radars are less capable in describing the shape of an object than that of LiDAR. Cameras are mainly used for capturing the information about traffic lights, traffic signs, and other objects. In general, the images collected by a camera are examined and partitioned to extract key features involving potential objects of interest. The extracted information is then compared with a feature library for the purpose of image recognition. However, the functionality of a camera is dramatically crippled by strong light or bad weather.

3) Planning: The planning module of an autonomous vehicle can be divided into three layers: 1) mission planning [34]; 2) behavior planning [35]; and 3) motion planning [36]. In most cases, they are conducted in the sequential order shown in Fig. 3.

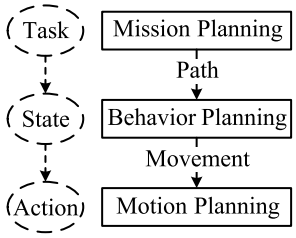

Fig. 3. Three layers of planning.

1) *Mission Planning:* Mission planning is also referred to as path planning or routing planning. It focuses on the task-level planning, such as the selection of a path between a starting point and an end point [37]. A given road system can be considered as a weighted directed diagram. This diagram contains plenty of information, such as the connectivity among the different roads, traffic rules, and the widths of the roads. This information contributes the *semantics* of an HD map mentioned in Section II-B1. As each directed edge in the diagram is weighted, the core idea of path planning for an autonomous vehicle is essentially the path search problem in a weighted directed diagram. In order to make a vehicle move from A to B, it is expected to obtain an optimal path, which is subject to several constraints, such as time, distance, and congestion.
2) *Behavior Planning:* Behavior planning is also called decision making. Since autonomous vehicles usually travel in a complicated environment, which is full of uncertainty and dynamics, challenges may come from: a) the degradation of the performance of the sensors and actuators, such as a snow-covered LiDAR and a skidding tire on wet ground; b) vehicles and pedestrians breaking the rules, or other objects, such as reckless animals and boxes falling off a truck; and c) unknown social conventions in unfamiliar areas, such as local festivals and gatherings. Therefore, behavior planning is introduced to make the appropriate decisions for the next move of the autonomous vehicle, according to the result of the mission planning and a wide variety of live information. For instance, behavior planning instructs the vehicle to follow or pass other vehicles, wait for or pass by pedestrians, etc. One approach to behavior planning is to use a complex finite-state machine (FSM), which contains a large number of actions [38], [39]. The FSM starts from an initial state and jumps to different states based on the variations of the driving scenario. The corresponding actions are passed to the motion planning.
3) *Motion planning:* Motion planning refers to the process of planning a series of consecutive actions. This series corresponds to a specific goal, such as acceleration and obstacle avoidance. In general, there are two important performance metrics for a motion planning algorithm: a) computational efficiency and b) integrity [37]. Computational efficiency refers to the processing speed of accomplishing a motion plan. The computational efficiency of a motion planning algorithm depends largely on the corresponding configuration space. The integrity

of a motion planning algorithm is described as follows. Provided a problem is solvable, the motion planning algorithm is able to find a solution in bounded time. For an unsolvable problem, the algorithm is capable of justifying its infeasibility. In the scenario of autonomous driving, the initial configuration of a motion planning algorithm usually contains the current states of the vehicle, including its position, linear velocity, angular velocity, etc. The target configuration is derived from the behavior planning. In practice, the movement of a vehicle always possesses certain restrictions, such as maximum steering angle, maximum acceleration, and maximum speed. These constraints are defined in the configuration space.

4) Control: When an autonomous vehicle completes its self-positioning and its perception of its surrounding environment, as well as its planning decision, it needs to transform the obtained series of action into controlled operations of the vehicle. In general, vehicle control consists of lateral control and longitudinal control [40]. Lateral control refers to the adjustment of the steering wheel and the tires' lateral force. Longitudinal control refers to the acceleration and braking of the vehicle.

In practice, the most common demands for control of an autonomous vehicle are acceleration, steering, and braking. The input of the control module is a series of path points. The role of the control module is to make the vehicle move along these path points to the greatest extent possible. A good control module should possess three features: 1) accuracy; 2) feasibility; and 3) stability. Feedback control is widely used in the field of automation control. The most typical feedback controller is the proportional–integral–derivative (PID) controller [41]. As a linear controller, ordinary PID controllers are widely used in industrial processes due to their simplicity. However, the application of a PID controller to autonomous vehicles faces the following challenge: the algorithms of a PID controller need to determine specific hyperparameters and their values [42]. For autonomous driving, the uncertainty of the external environment and the nonholonomic constraints of a vehicle make it difficult to find the appropriate hyperparameters and their corresponding optimal values.

5) Computing System: As the computing resources available to the onboard computing units are limited, it is difficult to deploy a large number of computation-intensive services on the vehicle. Edge computing is an effective way to address this problem. Zhang *et al.* [43] proposed a vehicular data analysis platform called OpenVDAP. The platform includes four main parts: 1) an onboard heterogeneous vehicle computing/communication unit (VCU); 2) an isolation-supported and security/privacy-aware vehicle operating system (EdgeOSV); 3) a driving data integrator (DDI); and 4) an edge-aware application library (LibvDAP). This platform is deployed on the autonomous vehicle to perform the calculations for the onboard applications. The service quality of the onboard applications and user experience is improved. Liu *et al.* [44] summarized the most advanced autonomous driving computing systems. There are seven performance indicators, nine key technologies, and 12 challenges.

III. Sensor Security

Autonomous vehicles are equipped with a variety of sensors, such as camera, GNSS/IMU, ultrasonic radar, millimeter-wave radar, and LiDAR. These sensors are responsible for collecting information about the positioning of the vehicle itself, its surrounding environment, etc.

El-Rewini *et al.* [45] presented a comprehensive review of potential cyber threats related to the sensing layer. Sensors of autonomous vehicles were classified as two categories: 1) vehicle dynamics sensors (e.g., TPMSs, magnetic encoders, and inertial sensors) and 2) environment sensors (e.g., LiDAR, ultrasonic sensors, cameras, radio detection and ranging systems, and GPS units). The authors also offered perspectives through existing countermeasures from literature and stressed the need for data-driven cybersecurity solutions.

Sensors are at the forefront of the field of autonomous driving. At present, most attacks against autonomous vehicles are related to sensors. Common attacks carried out against sensors inject misinformation or try to degrade the performance of the sensors by any means possible. As different sensors possess different operating principles, various types of attacks are used [26].

A. Camera

1) Role in Autonomous Driving: As computer vision assists autonomous vehicles to complete many perception tasks, the camera is the most basic vision sensor, and is indispensable for autonomous driving [46]. Cameras used by autonomous vehicles are mainly divided into three categories: 1) monocular cameras; 2) binocular cameras; and 3) multinocular cameras. The monocular camera is widely used in ADASs. However, there is a drawback to the use of a monocular camera. For a monocular camera with fixed resolution, a farther scene corresponds to a larger view, but it will be less clear. In contrast, a closer scene appears more clear. Although the binocular camera addresses the above problem of monocular camera, monocular cameras are used more than binocular cameras in autonomous driving at present. The main reasons are the expensive computational overhead of binocular camera algorithms and the shortage of space in an autonomous vehicle for such equipment.

2) Attacks and Countermeasures: In general, autonomous vehicles of Level 3/4 require the cooperation of multiple cameras for the perception of the surrounding environment, including pedestrians, lane lines, traffic signs, other vehicles, etc. In the task of traffic light recognition, if cameras capture a red light or a pedestrian, the vehicle should slow down or stop to avoid an accident. Hackers can place extra traffic lights or fake pedestrians to trigger a stop of the vehicle. In addition, a highlighted IR laser can also interfere with cameras, preventing the generation of effective images [47]. Attacks against camera and underlying computer vision algorithms of autonomous vehicles are common [47], [48].

Zhang *et al.* [49] proposed a framework based on three cameras to detect attacks against cameras. This framework uses

the information captured by the cameras to obtain different versions of depth maps (i.e., disparity).

Cao *et al.* [50] pointed out that all prior studies on autonomous driving systems only focused on camera or LiDAR-based autonomous driving perception alone. The authors studied the security of multisensor fusion (MSF)-based perception in autonomous driving. A novel attack pipeline was developed to attack all fusion sources simultaneously.

DiPalma *et al.* [51] developed an adversarial patch attack against camera-based obstacle detection. The adversarial patch with appropriate size and appearance is added to the back of a box truck. The experiments of the attack were conducted against an Apollo autonomous vehicle running in production-grade autonomous driving simulator LGSVL [52].

Kyrkou *et al.* [53] pointed out that advanced artificial intelligence and machine learning techniques play an vital role in proactive defense against attacks on autonomous vehicles' cameras. The authors developed a project called CARAMEL [54]. This project shows the use of AI/ML-based techniques in detection and possibly mitigation of dynamic cyber attacks on the camera system/data in autonomous driving. Both external attacks on camera sensor and direct attacks on camera sensor data were analyzed. Experiments were carried out on CARLA [55].

B. GNSS/IMU

1) Role in Autonomous Driving: GNSS/IMU is a real-time localization method in autonomous driving [56]. As a highly accurate localization method, GNSS-RTK is able to achieve centimeter-level position accuracy under dynamic measurement. Here, RTK stands for real-time kinematics. However, the frequency of location update is low, and the satellite signal can be easily blocked [57]. IMUs and odometers are used to accumulate displacement and direction variations for the purpose of compensation during the period between two consecutive positionings of the GNSS-RTK. Although the update frequency is high for the IMU and odometer, there are accumulated errors. Through the combination of GNSS and IMU, we can achieve real-time localization with low delay, high precision, and high frequency.

2) Attacks and Countermeasures: When a high-powered fake GPS signal transmitter is placed near an autonomous vehicle, the genuine GPS signal might be covered up. Thus, the localization of the autonomous vehicle is misled [58]. By combining two simple attack methods, GNSS signal jamming and spoofing, GNSS/IMU localization can be easily compromised [59].

Magiera and Katulski [60] proposed a spoofing detection method using phase delay measurement. This method uses multiple antennas to receive GPS signals of different qualities. Then, the accuracy and precision of the phase delay estimation are assessed.

In order to eliminate spoofing signals, Han *et al.* [61] constructed the subspace projection of the spoofing signals using the pseudocode characteristics of spoofing signals.

Dasgupta *et al.* [62] proposed a prediction-based spoofing attack detection scheme with the long short-term memory (LSTM) model. The distance between two consecutive locations of an autonomous vehicle is predicted by the LSTM model. Experiments were conducted with a real-world driving dataset called Comma2k19 [63].

Mit *et al.* [64] analyzed Tesla's Level 2 autonomous driving system under different GNSS spoofing scenarios. To examine various multiconstellation mitigation, GPS was spoofed and other constellations were jammed.

Dasgupta *et al.* [65] developed a deep reinforcement learning (RL)-based turn-by-turn GNSS spoofing attack detection using low-cost in-vehicle sensor data. The experiments were carried out with the Honda Research Institute Driving Data set [66].

Broumandan and Lachapelle [67] proposed a spoofing detection model based on consistency check between GNSS and IMU/odometer package. This model focuses on the utilization of inertial measurement units and vehicle odometer readings.

Song *et al.* [68] developed a credible navigation algorithm for GNSS attack detection using an auxiliary sensor system. A credible Kalman filter and measurement information given by the auxiliary sensor system are used to verify the credibility of the GNSS positioning result.

C. Ultrasonic Sensor

1) Role in Autonomous Driving: Ultrasonic sensors were first introduced into vehicles for automated parking assistance systems [69]. An ultrasonic sensor emits an ultrasonic signal in a certain direction through ultrasonic transmitting devices. A timer starts at the moment the signal is transmitted. The emitted ultrasonic signal is reflected back when it encounters obstacles during the transmission. When the reflected signal is received by the corresponding receiver, the timer stops. Based on the recorded time interval, the distance between the vehicle and the obstacle can be calculated.

2) Attacks and Countermeasures: Attacks threatening ultrasonic sensors mainly include spoofing attacks and jamming attacks.

Xu *et al.* [33] developed random spoofing, adaptive spoofing, and jamming attacks on ultrasonic sensors and validated these attacks on stand-alone sensors and moving vehicles.

Yan *et al.* [70] conducted an actual experiment with a spoofing attack in which an ultrasonic signal generated by hackers was introduced ([70, Sec. 5]). The generated signal is designed to reach the receiver of the vehicle earlier than the genuine signal expected to be reflected back.

Lim *et al.* [71] conducted an in-depth evaluation of vulnerabilities of ultrasonic sensor for autonomous vehicles. Several experimental attacks against ultrasonic sensor are launched.

Lou *et al.* [72] thoroughly studied the signal injection attacks and proposed a physical-layer defense system (SoundFence) to secure ultrasonic sensors in autonomous vehicles.

D. Millimeter-Wave Radar

1) Role in Autonomous Driving: Millimeter wave generally refers to an electromagnetic wave with a wavelength

of 1–10 mm. In most countries, vehicle-mounted millimeter-wave radar operates in the frequency bands of 24 and 77 GHz [73]. In addition, a few countries have adopted the frequency band of 60 GHz (e.g., Japan). Millimeter wave is able to work in rainy, foggy, and snowy weather conditions due to its strong penetrating ability.

2) Attacks and Countermeasures: If a hacker obtains the waveform parameters of a millimeter wave, a millimeter-wave radar at the same frequency band may be jammed [33]. Moreover, millimeter wave may also be subject to electromagnetic interference.

Yan *et al.* [70] conducted security experiments on the radar and autopilot system in Tesla Model S ([70, Sec. 6]). The experimental results showed that millimeter-wave radar of an autonomous vehicle suffers from electromagnetic jamming and spoofing. The authors also proposed that randomness should be introduced into control parameters, taking logic check, confidence priority, and attack detection system into consideration when designing a sensor data fusion strategy.

Kapoor *et al.* [74] proposed a spatiotemporal challenge–response (STCR) method. This method emits probing signals in multiple randomly selected directions at the same time. Then, the reflected signals are verified according to their directions of emission and arrival.

Digital radio-frequency memory (DRFM) [75] is a kind of microwave signal storage system, which is characterized by using a digital form to store the signals.

Guan *et al.* [76] proposed an anti-jamming method based on hash functions. The experimental results showed that the method is significantly effective in suppressing the echo interference.

Sun *et al.* [77] conducted an end-to-end security analysis of a millimeter-wave-based sensing system in autonomous vehicles. Practical physical layer attacks and defense strategies were implemented. Five real-world attack scenarios were constructed to spoof a victim autonomous vehicle.

E. LiDAR

1) Role in Autonomous Driving: LiDAR is currently the most important sensor for autonomous driving. The operating principle of LiDAR is to emit a laser beam and receive signals reflected back from a target. Several pieces of information related to the target can be obtained by comparing the outgoing and incoming signals, such as distance, azimuth, altitude, and even shape. LiDAR generates HD maps by capturing dense 3-D point cloud data from stationary and moving objects around itself. The advantages of LiDAR lie in its long detection range and accurate describing ability for 3-D information of objects.

2) Attacks and Countermeasures: Like the above-mentioned four sensors, LiDAR can also be easily interfered with. The main ways to attack LiDAR are the spoofing attack and the relay attack. A spoofing attack refers to injecting signals into the LiDAR receivers of the target vehicles, while the relay attack refers to using a transmitter and receiver to inject and receive the signals of the target vehicles, respectively.

Shin *et al.* [78] used a delay component to delay the LiDAR signals returned from a target vehicle. The delayed signals are emitted to the target vehicle by a malicious transmitter.

Cao *et al.* [79] showed two types of attacks: 1) an attack device placed at the roadside emits malicious laser pulses at passing autonomous vehicles and 2) an attack device carried by a vehicle emits malicious laser pulses at nearby victim vehicles.

Petit *et al.* [47] used two transceivers to relay LiDAR signals from the target vehicle to another vehicle at a different location.

Sun *et al.* [80] proposed CARLO to mitigate spoofing attacks on LiDAR. CARLO uses ignored occlusion patterns in the LiDAR point clouds as invariant physical features.

Changalvala and Malik [81] developed a 3-D quantization index modulation (QIM) data hiding technique for the purpose of securing the raw data from the LiDAR sensor. The experiments conducted on the KITTI object detection benchmark suite [82] showed that the proposed method was able to detect and localize insider data tampering attacks.

Yang *et al.* [83] proposed an adversarial attack against deep learning models, which perform object detection on raw 3-D points collected by a LiDAR sensor of an autonomous vehicle.

You *et al.* [84] developed a general methodology called 3-D temporal consistency check (3D-TC2). It takes advantage of spatiotemporal information from motion prediction to verify objects detected by 3-D object detectors.

F. Multisensor Cross-Validation

When observations from several different sensors are combined, there is a robust and comprehensive perception model for autonomous vehicles. In general, for the above five types of sensor, it is easy to attack an individual sensor. However, attacking all the sensors of an autonomous vehicle at the same time becomes more difficult. Currently, production autonomous driving systems predominantly adopt an MSF-based design, which, in principle, can be more robust against attacks under the assumption that not all fusion sources are (or can be) attacked at the same time [50]. Thus, it is expected that MSF technologies can effectively mitigate sensor attacks on autonomous vehicles. When the information coming from different sources is inconsistent, the vehicle might be under attack. For example, when GNSS/IMU and LiDAR yield different positioning results, at least one of the two systems might have been attacked. Besides, if a sensor system of an autonomous vehicle believes there is a traffic light, but HD Map indicates there is no traffic light at the same position, then in most cases the sensors of the vehicle are likely to have been attacked.

G. Sensor Failure

Onboard vehicle sensors may fail due to bad calibration, erroneous readings, physical or electrical failure, etc. Besides being caused by attacks, abnormal sensor readings may also be caused by failure. However, there is no standard or universally agreed definition for sensor failure [85].

Realpe *et al.* [86] proposed a system called the fault-tolerant perception paradigm for fault detection of sensors in autonomous vehicles. The system deals with possible sensor failure by defining a federated data fusion architecture.

Pous *et al.* [87] used analytical redundancy and a nonlinear transformation to generate residual signals for the detection of faulty sensors. The method uses statistical tools to optimally determine a threshold based on the characteristics of the signal, prior probabilities, and other information.

Byun *et al.* [88] proposed a fault diagnosis logic and signal restoration algorithm. The premise of this method is that only one sensor fails at any given time.

H. Actual Sensor Failure Versus Attacks

Both actual sensor failure and attacks might lead to wrong decisions in autonomous driving. Moreover, certain attacks are designed in an oversimplified and crude way. They simply aim to cause sensor failure. However, actual sensor failure and attacks against sensors are different.

In most cases, attacks against sensors tend to proceed stealthily. The tampering of sensor data is often mild and not obvious. The tampered sensor data just seems like the normal data. Besides, the expected attack effect is to fool the high-level algorithms by tampering the sensor data. On the contrary, actual sensor failure often results in obvious changes of sensor data, such as no readings for a significant time period, extremely high or low readings. For an MSF system such as the autonomous driving system, actual sensor failure can be easily noticed by multisensor cross-validation. In this case, safety measures can be taken timely. Thus, security and safety issues are likely to be prevented. On the contrary, as attacks are hard to be detected, both capacity-constrained artificial intelligence packages equipped with autonomous driving system and a negligent human driver will not be aware of an attack until serious incidents happen (e.g., a traffic accident). To the best of our knowledge, there is no literature concerning distinguishing between actual sensor failure and attacks. Researchers tend to study methodologies and techniques to discover and defense attacks. Actual sensor failure is left as hardware problems. Though actual sensor failure also leads to abnormal sensor data, it is often neglected by researchers. The effect of actual sensor failure is just treated equally as that of attacks. Thereby, researchers just try to mitigate the consequences of both actual sensor failure and attacks, such as [89] and [90].

I. Drawbacks of Existing Protection Methods

At present, the research on sensor attacks on autonomous vehicle is still at an early stage. The methods of protection against sensor attacks mainly focus on a single type of sensor. Little attention has been paid to detection methods for the cases of multiple types of sensor being attacked. On the whole, there is no systematic theory or architecture for the detection of and defense against attacks. In addition, most existing protection methods focus on the detection of attacks. For an identified attack, there are no recovery methods for sensor data, which are able to work in an intrusion-tolerant manner.

IV. Operating System Security

An autonomous driving system integrates multiple software modules, such as localization, perception, planning, and control. These modules need to meet certain real-time requirements. Therefore, autonomous driving requires an operating system to manage these modules. The operating system mainly provides the functions of communication and resource allocation among the modules. Next, we discuss the security of the operating system. The sensors of an autonomous vehicle continuously generate data during their operation. The processing of data generated by each sensor imposes strong real-time requirements on the operating system. Due to the strong connections among the modules in the autonomous driving system, effective communication and resource allocation among the modules become challenging.

A. Early Mobile Robot Operating Systems

Before autonomous driving, there were mainly three popular mobile robot operating systems.

1) Miro: Miro is an object-oriented robot middleware. Technically, Miro implements an object-oriented design by adopting the common object request broker architecture (CORBA) standard [91].

2) URBI: URBI is a universal robotic body interface based on a client/server architecture [92]. URBI does not provide a graphical programming interface.

3) OpenRDK: OpenRDK is a modular management framework for designing distributed robot systems [93]. OpenRDK is implemented with C++.

These three operating systems mainly provide a software component management framework for mobile robots. Since these operating systems lack software libraries and visual debugging tools, they are not suitable for autonomous vehicles. In fact, they are not used by any autonomous vehicles. Initially, the operating system of most autonomous vehicles was basically developed based on ROS.

B. ROS

ROS is a powerful and flexible robot programming framework. It is a distributed multiprocessing framework based on messaging. Many key components of autonomous driving are implemented on ROS, such as quaternion-based coordinate transformation [94], a robotic 3-D mapping framework [95], and the positioning algorithm SLAM [96]. The message mechanism of ROS enables a modular design based on software functions. Each module is able to read and distribute messages.

C. Security of ROS

Attacks on sensors are external attacks that do not require access to the autonomous vehicle's operating system. Internal attacks involve hacking into the autonomous vehicle's operating system. The autonomous vehicle's operating systems implemented based on ROS have a common security issue: ROS does not provides authentication for messaging and node creation [97]. There are mainly two types of attack [4]: 1) a hijacked ROS node is able to continuously generate and distribute messages. This kind of malicious behavior might make

the system run out of memory (OOM). Then, the autonomous vehicle's operating system would start to close ROS node processes. This would result in a crash of the operating system and 2) messages sent by a hijacked topic or service of ROS may be tampered with or forged, thus leading to abnormal behavior of the operating system.

The first attack is rooted in the fact that ROS has no isolation mechanism; thus, an ROS node is able to access system resources without any restriction. The source of the second attack resides in the fact that the messaging among nodes is not encrypted; thus, attackers are able to obtain the message content readily [98].

SROS [99] is a set of security enhancements for ROS. There are transport layer security (TLS) support for communication within ROS, the use of x.509 certificate permitting chains of trust, definable namespace globbing for ROS node restrictions and permitted roles, covenant user-space tooling for the autogeneration of node key pairs, audit ROS networks, and construct/train access control policies. Zhang *et al.* [100] proposed an access control framework named AC4AV for autonomous driving vehicles. Different access control models are developed to protect in-vehicle data in real-time data and historical data.

Apollo 3.5 and later versions replace the original ROS middleware and use the Apollo Cyber role-based trust (RT) middleware instead [101]. Unlike ROS, there is no master node in Cyber RT. The entire network topology of Cyber RT is divided into different domains. When a new node joins the network, it sends broadcast messages to other nodes in the domain with a real-time publish subscribe (RTPS) protocol [102].

Xu *et al.* [103] deployed a data-driven prediction architecture for autonomous driving on the Apollo platform. The architecture enables rapid and efficient deployment of Apollo's prediction technologies across different regions.

D. Security Enhancement of ROS2

A DDS [104] was first applied in the U.S. Navy to handle the compatibility problem of a large number of software upgrades in the complex network environment of its ships [105]. It has become a standard solution for data publish/subscribe in distributed real-time systems. An autonomous vehicle's operating system needs to establish a universal, high speed, and efficient DDS framework across multiple cores, multiple CPUs, and multiple boards. DDS is able to ensure a real-time, efficient, and flexible distribution of data and meets the needs of various distributed real-time communication applications. The security standard for DDS implements three-way handshakes, which contains three messages: 1) HandshakeRequest; 2) HandshakeReply; and 3) HandshakeFinal [106].

The DDS security specification defines five service plugin interfaces (SPIS) to increase security [107].

1) *Authentication Service Plugin:* This is central to the entire SPI architecture. It provides methods to verify the identity of an application or user that invokes operations on DDS.
2) *Access Control Service Plugin:* This defines and enforces restrictions on the DDS-related capabilities of a domain participant.
3) *Cryptographic Service Plugin:* This handles all cryptography-related operations, including encryption, decryption, hashing, signature, etc.
4) *Logging Service Plugin:* This provides for the auditing of DDS security-related events.
5) *Data Tagging Service Plugin:* This tags specific DDS security-related actions performed by the users, providing the ability to add tags to data samples.

Unlike Apollo, Autoware [108] is currently developed based on ROS2 [109]. ROS2 has made significant improvements to the original ROS framework. It uses an advanced distributed architecture, rather than the original master–slave structure. ROS2 adopts DDS as its messaging model. The DDS security extensions are used to protect the data during transmission [110]. The adoption of DDS improves the reliability and real-time performance of multirobot collaboration.

DDS is an industry standard implemented by many companies, such as RTI implementation Connext [111], eProsima implementation Fast DDS [112], and ADLINK implementation DDS [113]. There are many aspects to consider when choosing a DDS implementation, such as protocol legality and whether it is cross-platform. In order to prevent ROS2 from depending on a specific DDS program, ROS2 supports multiple implementations. Morita and Matsubara [114] proposed a dynamic binding mechanism, which is able to choose an appropriate DDS implementation.

Compared with ROS, ROS2 is enhanced in the following three aspects [115].

1) *Real Time:* DDS has a variety of transport configurations, such as deadline, fault tolerance, and reliability. It brings real-time support to ROS2.
2) *Continuity:* Although ROS has the concept of a data queue, it still has great limitations. For instance, subscribers cannot receive data before joining the network. But DDS can provide data history service for ROS. Even a newly added node can obtain all the previously released data.
3) *Reliability:* Based on the DDS reliability configuration, users can choose the performance mode (BEST_EFFORT) or the stable mode (RELIABLE) according to their demands.

At present, the security of ROS2 is highly dependent on the security of DDS [106]. The implementation of ROS2 only employs the first three SPIS of DDS mentioned above.

1) *Builtin Authentication Plugin (Called "DDS: Auth: PKI-DH"):* This plugin uses a verified PKI. It requires each participant to have a public key, a private key, and an x.509 certificate.
2) *Builtin Access Control Plugin (Called "DDS: Access: Permission"):* This plugin also uses a PKI. It requires two signed XML documents per domain participant: a) a governance file and b) a permissions file.
3) *Builtin Cryptographic Plugin (Called "DDS: Crypto:AES-GCM-GMAC"):* It provides authenticated

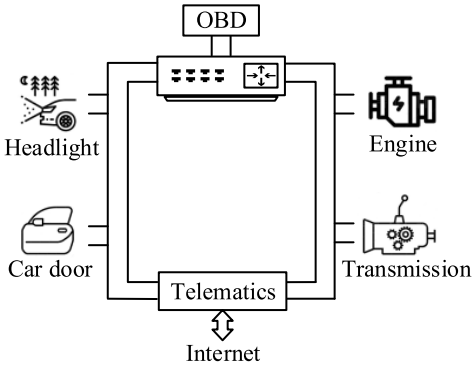

Fig. 4. CAN bus network.

encryption using advanced encryption standard (AES) in Galois counter mode (GCM), namely, AES-GCM.

The main reason why ROS2 uses built-in plugins instead of other plugins is to allow all compatible DDS implementations to be interoperable with ROS2. Thus, the security features of ROS2 are able to work with all vendors with minimal effort.

E. Drawbacks of ROS2

ROS2 lacks certain vital mechanisms. Here, are two examples: 1) *secure OTA update [116]:* this establishes a connection between a background server of the vehicle manufacturer and an autonomous vehicle by WiFi. Update packages are downloaded from a server to update the local software of the vehicle. If the OTA is compromised by hackers, the security of autonomous vehicles will be affected and 2) *secure key exchange [117]:* current solutions for a communication channel for key exchange between remote listeners and talkers are not sufficiently secure. Thus, they are vulnerable to key interception attacks.

V. Control System Security

Various mechanical components and digital devices in autonomous vehicles are controlled by ECUs. The communication among different ECUs in a vehicle is conducted by a digital bus.

A. CAN

CAN is the main bus protocol of the in-car electronic network [118]. It has the advantages of stability and reliability, strong real-time performance, strong anti-jamming ability, and long transmission distance. A CAN bus adopts differential signal transmission. In general, its normal communication only needs two signal lines: 1) CAN-H and 2) CAN-L. The two possess opposed characteristics to avoid external electromagnetic interference and radiation [119]. In a CAN, a node can initiate communication to other nodes at any time. There is no master–slave relationship between the nodes. However, the right to use the bus is in accordance with node priorities. An autonomous vehicle often adds several telematics nodes in the CAN bus network [120]. As shown in Fig. 4, these nodes are connected to the CAN bus in order to facilitate remote control, remote upgrade, and other functions. Hackers can hack into the CAN bus network through the onboard diagnostics (OBD) port.

B. Vulnerabilities of CAN and Attack Methods

Currently, as the CAN bus has no authentication or access control, it is easily hijacked by hackers [121]. There have been many car network attacks against the CAN bus. Miller and Valasek [122] used system vulnerabilities to remotely control a Jeep's multimedia system. Then, they attacked the V850 controller and modified its firmware to obtain permission to remotely send commands to the CAN bus for the purpose of controlling the power system and braking system. This issue caused a recall of 1.4 million vehicles. Greenberg [123] also attacked a Jeep's CAN bus and successfully controlled the steering, braking, acceleration, etc.

Generally speaking, it is difficult to get into the CAN bus itself. However, the entertainment system and the OBD-II port of the maintenance system are connected to the CAN bus. These connections expose possible attack paths to the CAN bus. Five popular attack paths are as follows.

1) OBD-II Invasion: OBD-II improves OBD in terms of diagnostic functions and standardization. The OBD-II port is mainly used to access vehicle status. During vehicle maintenance, technicians use the detection software (e.g., Ford's NGS, Nissan's Consult II, and Toyota's Diagnostic Tester) developed by vehicle vendors to manipulate the OBD-II port and examine the vehicle. Since the OBD-II port is connected to the CAN bus, hackers who have access to the detection software can easily intercept information on the CAN bus and control the vehicle [124].

2) Invasion of Chargers for Electric Vehicles: Charging equipment is an essential component of an electric vehicle. The charging equipment also connects to the CAN bus. As the charging equipment of an electric vehicle communicates with an external charging pile, hackers have the opportunity to invade the CAN bus from the external charging pile [125].

3) CD Player Invasion: In general, a media player is connected to the CAN bus. Hackers can encode attack codes into a music CD. When the CD is played, the malicious codes invade the CAN bus from the CD player. Hence, the hackers are able to control the CAN bus [20].

4) TPMS Invasion: TPMS stands for the tire pressure monitoring system. For the attack path, hackers inject attack codes into the TPMS. When the TPMS detects a specific value of tire pressure, the malicious codes are activated to attack the vehicle [126].

5) Bluetooth Invasion: Autonomous vehicles support Bluetooth connections to other electronic devices (e.g., smartphones, personal digital assistants, and laptops). Malicious programs on smartphones are able to communicate with the CAN bus by the Bluetooth connection [127].

As the CAN bus lacks authentication, a CAN frame only indicates its destination. There is no information of the source of the message. As a result, malicious information can be regarded as valid information as long as the message format is correct. Based on this issue, the security protection methods for CAN bus fall into two categories: 1) those based on encryption/authentication and 2) those based on intrusion detection.

C. Protection Methods for CAN Bus

1) Methods Based on Encryption/Authentication: These methods mainly conduct authentication for messages and ECUs or encrypt messages to ensure the security of the CAN bus. As the CAN bus lacks encryption schemes and the frame size is small, this kind of method often requires adding hardware to the ECUs or upgrading the existing firmware.

Groll and Ruland [128] employed a key distribution center in the vehicle network to divide the vehicle network into different areas. Different keys are assigned to these areas for communication.

To prevent attackers from sniffing and tampering with the ECU codes, Yu *et al.* [129] used a Markov decision process to model the interaction between the attacker and the system and encrypted the storage system of the onboard ECU.

Murvay and Groza [130] implemented a method for identifying the sources of the messages, based on an analysis of the frames on the bus.

Wang and Sawhney [131] proposed a framework named Vecure to protect the CAN bus of vehicles. This framework uses the structure of a trust group to strengthen access control and prevent false messages from entering the CAN bus network.

Woo *et al.* [132] sent attack messages to the CAN bus network remotely through Bluetooth and OBD-II. For this attack, they presented a lightweight message encryption method based on the advanced encryption standard-128 (AES-128) algorithm.

2) Methods Based on Intrusion Detection: Methods based on intrusion detection focus on establishing a detection model by analyzing the time series, frequency, and other characteristics of the messages. This kind of method introduces less overhead than using encryption and authentication. However, these methods require a more comprehensive understanding of a vehicle's CAN protocols. In addition, the false alarm rate of these methods is higher than that of the other kind.

Ning *et al.* [119] used a local outlier factor (LOF) to identify attacks and detect intrusions in automotive networks. Data packets transmitted by different ECUs on the CAN bus produce distinct voltage waveforms.

Song *et al.* [133] proposed a lightweight intrusion detection algorithm for a CAN bus based on an analysis of the time intervals of the CAN messages. This proposal is rooted in the periodicity of the CAN messages.

Taylor *et al.* [134] proposed an anomaly detection method based on the statistics of the traffic in the vehicle network. This method is able to detect injection attacks aimed at messages. However, it cannot detect attacks aimed at aperiodic messages.

Cho and Shin [135] proposed a clock-based intrusion detection system, which analyzes the clock offsets of the vehicle-mounted message timestamps to detect various attack scenarios.

Marchetti and Stabili [136] constructed multiple models of normal ID sequences of the collected messages based on the transition matrix group.

Taylor *et al.* [137] proposed a learning model based on an LSTM network to detect message sequences in the CAN bus. The learning model predicts the next data word from each sender on the bus.

Kang and Kang [138] studied an intrusion detection system using a deep neural network (DNN). The system employs probability-based feature vectors extracted from messages in a vehicle-mounted network to train the parameters of the DNN.

Markovitz and Wool [139] developed a greedy algorithm to split messages into different fields. Then, a semantically aware anomaly detection system is built based on the field classification.

D. CAN FD

CAN FD was initially introduced as a specification [140] of BOSCH [141] in 2012. Then, it was formally presented in [142]. CAN FD is able to perform standard CAN communication. It shares the physical layer with the CAN as defined in the BOSCH CAN specification [143]. CAN FD can be considered as a protocol, which provides efficient distributed real-time control with a high level of security. Safe data transfer, cogent error detection, signaling, and self-checking are implemented in the CAN FD node. Though CAN FD is considered to be the next-generation in-vehicle network protocol, it has some security vulnerabilities suffered by CAN [144]. When a CAN data frame is broadcasted, the confidentiality and authentication are not guaranteed. CAN FD is also vulnerable to the above problem and suffers from eavesdropping and replay attacks.

Woo *et al.* [144] proposed a seven-phase security architecture for in-vehicle CAN FD. Based on the analysis of attack models, the proposed architecture contains long-term symmetric key exchange, authenticated key exchange, encryption/authentication of CAN FD data frames, etc.

Xie *et al.* [145] pointed out that CAN FD lacks a security authentication mechanism and is vulnerable to masquerade attacks. The authors developed a two-stage security enhancement for real-time parallel in-vehicle applications.

Xie *et al.* [146] proposed a security-aware obfuscated priority assignment approach for CAN FD messages.

Xie *et al.* [147] developed a security enhancement method for independent in-vehicle CAN FD messages. The proposed method is able to dynamically adjust the MAC size of an independent message.

Yu and Wang [148] pointed out that unauthorized devices are able to access CAN FD by embedding external intruding devices to in-vehicle networks.

Xie *et al.* [149] proposed an AUTOSAR-compliant system model, which considers both time and security constraint. Here, AUTOSAR stands for automotive open system architecture [150]. The model is defined as the basis for the design space exploration (DSE) method of CAN FD.

Xiao *et al.* [151] pointed out that a key security mechanism message authentication between ECUs for countering message spoofing and replay attack is crucial to the AUTOSAR-compliant system proposed in [149]. As the session key establishment with AUTOSAR compliance was not well addressed, the authors developed an AUTOSAR-compliant key management architecture.

Agrawal *et al.* [152] developed a security architecture for the communication between ECUs on different channels through gateway ECU (GECU).

E. Drawbacks of Existing Protection Methods

1) CAN: Most methods based on encryption/authentication require an update of the current CAN hardware. Moreover, these algorithms introduce extra computation into the CAN bus. This may affect the real-time performance of the CAN bus. Most existing methods based on intrusion detection can only be applied to a limited number of intrusion scenarios. Moreover, the actual performance of these methods is still unsatisfactory in terms of the false positive rate. In summary, both these types of protection methods contain complicated algorithms and introduce significant computational costs. Thus, the real-time requirements of a CAN bus are hardly met.

2) CAN FD: Though CAN FD is superior to CAN in terms of data payload size and bandwidth consumption, security is not well addressed for CAN FD. All attacks, which are possible to CAN, are also applicable for CAN FD [152]. With the increasing number of external intruding devices, the real-time performance of security enhancement built on topology construction/optimization is compromised. Moreover, popular security measures for CAN FD are based on encryption/authentication and intrusion detection methods, as well as for CAN. The design and implementation of these techniques are seriously confined by the real-time requirements for autonomous driving systems.

VI. V2X Communication Security

When an autonomous vehicle is on the road, it becomes part of the IoV. V2X is a catch-all term for the communication mechanisms of the IoV. As mentioned in Section I, these mechanisms usually include V2V, V2I, V2P, and V2N. A vehicle can obtain a series of traffic information (e.g., real-time traffic status, pedestrians, and status of surrounding vehicles) with V2X. Protecting the security of V2X communication is an important domain of autonomous driving. In this section, we discuss the potential security risks of V2X and corresponding solutions.

A. V2X Communication

The four kinds of communication in the V2X are shown in Fig. 5: V2V, V2I, V2P, and V2N. In V2V, the most common application scenarios are urban streets and highways, where vehicles send data to each other for information sharing. This information includes the vehicle's speed, direction of motion, acceleration, braking, relative position, steering, etc. By predicting the driving behavior of other vehicles, a vehicle is able to take safety measures in advance. In V2I, vehicle-mounted devices communicate with the infrastructure point roadside units (RSUs). The RSUs obtain information about nearby vehicles and publish real-time information on Internet portals. In V2P, vehicles identify the behavior of nearby pedestrians with multiple sensors. When necessary, warnings can be issued with lights and the horn. It is expected that pedestrians will then become aware of the potential danger. In V2N, vehicle-mounted devices communicate with cloud servers to exchange information. The cloud stores and analyzes the uploaded data to provide various services to the vehicles, such as navigation, remote monitoring, emergency assistance, and in-car entertainment.

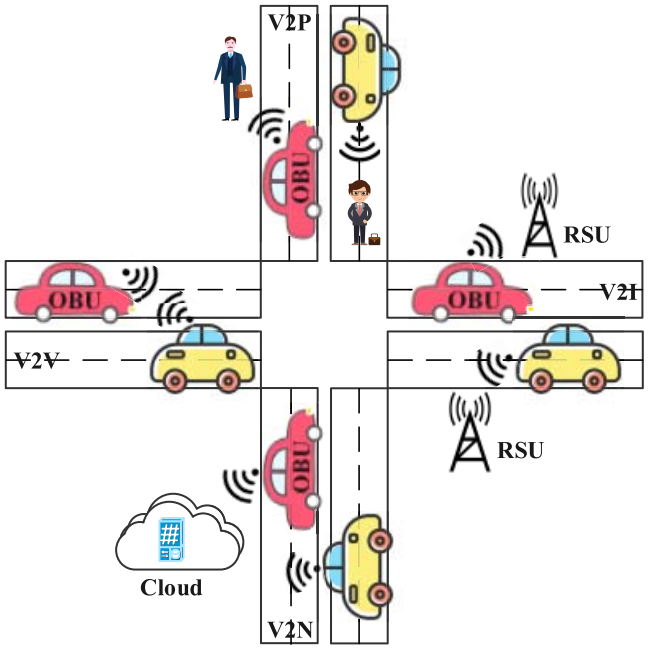

Fig. 5. V2X communication network.

B. V2X Communication Attacks and Solutions

Hasrouny *et al.* [153] presented a classification of attacks on V2X based on the compromised services. The attacks are classified into four groups: 1) authenticity/identification; 2) availability; 3) data integrity; and 4) confidentiality. Here, we conduct an in-depth study based on this classification and review several notable publications. Representative studies of these four categories are summarized in Table I.

1) Authenticity/Identification Attacks and Countermeasures:

1) *Sybil Attack:* In a vehicular ad hoc network (VANET), a vehicle joining the network becomes a wireless node. Since a node may join and leave a VANET freely, data are backed up among multiple nodes to enhance its availability to the network. An attacker may use a single malicious node to masquerade multiple identities, data being backed up in the same malicious node. Similarly, malicious messages can be propagated to other nodes by the same malicious node with multiple identities. For example, an attacker may propagate a fake traffic scene to several nodes. When another normal node in the network receives the fake traffic scene from those nodes, the normal node may modify its driving route. This may lead to a traffic accident [189]. Park *et al.* [154] proposed a detection method based on timestamp series. The method does not need a special vehicular PKI to authenticate a vehicle. Li *et al.* [155] introduced a public-key encryption model of pseudonym generation. This scheme allows a legitimate third party to obtain the real ID of a vehicle for identity authentication. Yao *et al.* [156] proposed a method for detecting Sybil attacks based on vehicular voiceprints. Received signal strength indicator (RSSI) time series are used as vehicle-mounted speech to measure

TABLE I
ATTACKS ON V2X

Attack Classification	Attack	Attack Behavior
Authenticity/Identification	Sybil [154] [155] [156] [157]	Create multiple vehicles with the same identity
	Key or Certificate Replication [158] [159]	Steal certificates/keys
	GNSS Spoofing [160] [161]	Provide false location information
	Timing [162] [163]	Introduce transmission delay
Availability	DoS [164] [165] [166] [167]	Send numerous useless requests to compromise service availability
	DDoS [168]	Launch multiple DoS attacks from different nodes
	Spamming [169]	Send spam messages to consume bandwidth
	Flooding [170]	Broadcast false messages in the network
	Wormhole [171] [172]	Send packets over private channels
	Blackhole [173] [174] [175]	Discard relayed packets
	Malware [176]	Inject viruses into software by insiders
	Jamming [177] [178]	Transmit interference signals to communication channels
	Broadcast Tampering [179] [180]	Tamper with security messages in communication channels
Data Integrity	Masquerading [181]	Use a valid identity to hide
	Replay [182] [183]	Send previous messages repeatedly
	Illusion [184] [185]	Create fake traffic messages
	Message Alteration [186]	Modify, add, and discard the data packets
Confidentiality	Traffic Analysis [187]	Monitor the network and analyze packets to infer sensitive information
	Eavesdropping [188]	Obtain confidential data by unauthorized access

the similarity of the received series. Feng *et al.* [157] proposed an event-based reputation system (EBRS) to detect a Sybil attack on a VANET.

2) *Key or Certificate Replication Attack:* An attacker sniffers the network to obtain a certificate/key. The obtained credentials are then sent to an authentication server to declare a legal identity [190]. Azees *et al.* [191] proposed an efficient anonymous authentication scheme with conditional privacy preserving (EAAP) to deal with key or certificate replication attacks on a VANET. Oulhaci *et al.* [159] proposed a distributed vehicle authentication architecture based on public keys.
3) *GNSS Spoofing Attack:* In a VANET, accurate and reliable location information is crucial to the operation of the whole network. An interference system designed by hackers generates false navigation signals, which mislead the GNSS navigation of a vehicle. As the planning of autonomous vehicles is highly dependent on the sensor data, this attack is quite serious for an autonomous vehicle [192]. Curran and Broumendan [160] proposed a method, which uses uncalibrated low-cost IMUs to detect GNSS spoofing attacks. However, a subsequent study [193] showed that ultrasonic pulses can stimulate certain microelectro mechanical systems (MEMS) sensors. This may cause IMUs to generate false measurements. Wang *et al.* [161] proposed a method based on edge computing to reconstruct unavailable and untrustworthy GPS signals. The implementation of this method does not require the vehicles to carry any additional equipment (e.g., antenna and receiver).
4) *Timing Attack:* The timing attack is to delay the transmission of messages with high real-time requirements. As most messages with high real-time requirements are critical to the operation of a vehicle and the whole VANET, a malicious node in the network, which introduces abnormal latency to certain messages, is of great harm [194]. Chuang and Lee [162] proposed a decentralized lightweight authentication framework called the trust-extended authentication mechanism (TEAM). Arsalan and Rehman [163] proposed a protocol timing attack prevention (TAP) method based on a software-defined network (SDN) [195], referred to as data networking (NDN) [196], to address the problem of the timing attack on a VANET.

2) Availability Attacks and Countermeasures:

1) *Denial of Service (DoS) Attack:* A DoS attack aims to exhaust the resources of a VANET by sending a large number of useless requests. In this case, normal requests from valid users cannot get served. This type of attack can be launched by malicious nodes inside or outside the network. When the network is filled with artificial malicious information, legitimate network nodes [e.g., onboard units (OBUs) and RSUs] are unable to work normally due to the scarcity of resources [197]. An enhanced version of the DoS attack is the distributed DoS (DDoS) attack. An attacker can control a large number of victim nodes to perform many DoS attacks on a VANET. These victim nodes are called zombie nodes. There are two scenarios for the DDoS attack on a VANET [168]: a) DDoS in V2V communication: Zombie nodes send message requests to a victim vehicle from different locations and time slots. For different types of nodes, the attacker can change the time slots and the content of the message requests. The attacker aims to overload the victim vehicle and bring down the network. As a result, the victim cannot access network resources and b) DDoS in V2I communication: attacks are launched from vehicles in different locations and the target is the RSUs. When the RSUs are overloaded, they are unable to respond to valid requests from normal nodes. Perrig *et al.* [164] proposed a timed efficient stream loss-tolerant authentication (TESLA) model. However, TESLA is vulnerable to memory-based DoS attacks. To address this problem, Studer *et al.* [165] proposed an effective authentication model for broadcast messages using symmetric cryptography and a delayed key. This model is called TESLA++, which is considered to be an improved version of TESLA. The advantage of TESLA++ is

the prevention of memory-based and computation-based DoS attacks. Liu *et al.* [166] designed a puzzle-based co-authentication (PCA) scheme. Jie *et al.* [167] proposed a mechanism to detect and filter malicious messages in a VANET by introducing port hopping [198] and a singular linear space [199].

2) *Spamming Attack:* Spamming attack is a type of DoS attack. In this type of attack, a large amount of spam is sent over the network to consume bandwidth, thereby increasing transmission delay on VANET [200]. Due to the lack of centralized management of the transmission medium, spamming control becomes considerably difficult. Malla and Sahu [169] proposed a redundancy elimination mechanism consisting of a rate decreasing algorithm and a state transition mechanism.
3) *Flooding Attack:* Flooding attack is also a type of DoS attack. The attacker broadcasts fake messages to the VANET through malicious nodes, which can consume a lot of resources and reduce the throughput of the network. In this case, the network stops service for a certain time period [201]. Faghihniya *et al.* [170] proposed a method, called the bus ad hoc on-demand distance vector (B-AODV) protocol, for detecting the route request (RREQ) flooding attack.
4) *Wormhole Attack:* In VANET, a wormhole attack involves an attack in which the malicious nodes use the private channel already established in the network to transmit information that has been stolen from the network to another location in the network instead of transmitting it via a normal network connection. A malicious node can make any possible attack, such as packet dropping, data tampering, traffic analysis, etc., on the data passing through the wormhole tunnel [202]. Safi *et al.* [171] used a packet leash and an authentication method called HEAP. Ali *et al.* [172] used the public key cryptosystem RSA and symmetric key encryption technology to broadcast messages securely.
5) *Blackhole Attack:* The blackhole attack is a conventional attack against the availability of VANET. After receiving the routing request packet, the malicious nodes in the network will claim to be the nearest nodes with low latency to the destination node, and thus, many nodes will choose them as the next-hop node for data packet forwarding. In the stage of data transmission, the malicious nodes usually directly discard the data packet without forwarding it. As a result, packet loss will occur in VANET [203]. Daeinabi and Rahbar [173] proposed an algorithm for detecting malicious vehicles that drop packets and isolate them from the normal vehicles. Baiad *et al.* [174] proposed a cross-layer cooperative blackhole attack detection scheme that consists of three main layers of defense. Abdulkader *et al.* [175] proposed a routing protocol called lifetime improving ad hoc on-demand distance vector (LI-AODV) to deal with the blackhole attack in VANET.
6) *Malware Attack:* In a malware attack, when OBUs and RSUs need patches or software updates, it is possible that malware, such as computer viruses, can disturb the operation of the network [204]. This type of attacker is usually a malicious insider rather than an outsider. Such an attack can be mitigated by using anti-malware or firewalls [176].
7) *Jamming Attack:* In a jamming attack, a moving vehicle is used as a node. The nodes communicate with each other by transmitting RF signals. However, due to the low reliability of mobile computing and the high scalability of the system in a wireless environment, attackers can launch high-power interference signals to the communication channel, causing the node to reduce or even lose the ability to receive data packets [205]. Mokdad *et al.* [177] proposed a new algorithm, called DJAVAN, to detect interference attacks in VANET. Karagiannis and Argyriou [178] proposed an unsupervised learning method to detect jamming attacks on vehicle communication. Benslimane and Nguyen-Minh [206] proposed an analytical jamming model, which is able to determine thresholds more accurately in threshold-based detection methods.
8) *Broadcast Tampering Attack:* In this type of attack, by injecting false security information into the network or tampering with the broadcast security messages, attackers force the legitimate vehicles to make choices that are not good for themselves, which might cause traffic accidents or increase the traffic flow on a certain road [207]. Wasef *et al.* [180] first described that PKI is a viable mechanism to protect VANET. He and Zhu [179] proposed a lightweight and efficient broadcast authentication scheme, which mainly adopted a one-way hash chain and group key update technology.

3) Data Integrity Attacks and Countermeasures:

1) *Masquerading Attack:* In this type of attack, attackers use forged identities to gain informal access to the network. In this way, they can alter or discard data packets transmitted in VANET. An example of this type of attack is a malicious node disguising itself as an emergency vehicle to force other vehicles to slow down or stop [202], [208]. Malhi and Batra [181] proposed a framework that uses genetic algorithms to detect and prevent masquerading attacks in VANET.
2) *Replay Attack:* In this type of attack, malicious vehicles repeatedly send messages from a certain time period in the past to other vehicles, causing them to be cheated and thereby, achieving the purpose of traffic jams. For example, a malicious vehicle saves messages about a traffic accident from a certain time period in the past and uses it to deceive other vehicles after the message expires [209]. Li and Song [182] evaluated the trustworthiness of traffic data and vehicle nodes and proposed an anti-resistant trust (ART) management scheme. Alazzawi *et al.* [183] proposed a scheme to deal with the replay attack in VANET. The scheme consists of six stages. Compared to the previous ID-based schemes [210], [211], the overall communication overhead of this scheme is lower.
3) *Illusion Attack:* In an illusion attack, an attacker manages to fake sensor readings on their vehicle to create fake traffic messages and broadcasts them to the neighboring

TABLE II
POPULAR SIMULATORS FOR NETWORKING AND COMMUNICATION IN AUTONOMOUS DRIVING

Type	Simulator	Language	Platform	Open source	Reference
Network Simulator	NS-2 [217]	C++, OTCL	Cygwin, Linux, MacOS	✓	[154] [156] [162] [165] [166] [180] [186]
	NS-3 [218]	C++, Python	Cygwin, Linux, MacOS	✓	[177]
	OMNeT++ [219]	C++, NED	Windows, Linux, MacOS	✓	[175]
	GloMoSim [220]	C, Java	Windows, Linux	✓	[171] [182]
Traffic Simulator	SUMO [221]	C++, XML	Windows, Linux, MacOS	✓	[163]
	VanetMobiSim [222]	Java, XML	Windows, Linux	✓	[174]
	TraNS [223]	C++	Linux	✓	[187]
	MATLAB [224]	MATLAB	Windows, Linux, MacOS	✗	[159] [174] [181]

nodes to cause traffic jams [212]. Lo and Tsai [184] used a plausibility network checking module and a rule database to verify the credibility of the message, mainly by checking whether the timestamp, speed, and other element fields of the given message conform to the corresponding predefined ruleset of the rule database. Zacharias and Fröschle [185] proposed a framework called the misbehavior detection system (MDS) to detect an illusion attack in VANET.

4) *Message Alteration Attack:* In this type of attack, the attacker alters the data packets in the network by adding, deleting, or discarding the data, resulting in the data integrity being broken [213]. Zhu *et al.* [186] divided the network into multiple domains, in which an RSU is responsible for allocating group private keys to localize the management of vehicles.

4) Confidentiality Attacks and Countermeasures:

1) *Traffic Analysis Attack:* In this type of attack, the attacker analyzes the traffic messages in V2X communication, extracts and collects as much information as possible that is beneficial to them (e.g., location of the vehicle and driving path of the vehicle), and induces bad behavior in other vehicles, which violates the data confidentiality in VANET [214]. Cencioni and Di Pietro [187] proposed a communication protocol called the V2I privacy enforcement protocol (VIPER) to deal with traffic analysis attacks in VANET. In order to prevent the attacker from learning the identity of the message sender from the message field, VIPER uses universal reencryption [215] to encrypt each message.
2) *Eavesdropping Attack:* Due to the broadcast nature of wireless communication of VANET, the communication among vehicles might be eavesdropped by illegal users. Eavesdropping attack is a common attack against confidentiality that is usually launched at the network layer. In this type of attack, the attacker obtains confidential data, such as the location data used to track a vehicle, for their own purposes [216]. Dai *et al.* [188] proposed a security framework based on indirect reciprocity. The framework assigns a scalar reputation to each vehicle node in VANET and this is used for estimating how dangerous each node is to the VANET.

C. V2X Communication Simulators

The research of V2X communication security requires powerful experimental support. Since experiments in a real environment consume manpower and other material resources, excessive experiments may be a waste of time for immature autonomous driving technologies. Generally speaking, two kinds of simulators are involved: 1) network simulator and 2) traffic simulator. Network simulators are used to test the performance of network protocols and applications, while traffic simulators are used to generate vehicle trajectories. Table II summarizes popular simulators for the research of V2X communication.

D. Drawbacks of Existing Countermeasures

Most existing countermeasures against V2X attacks require certain authentication schemes. As V2X related devices have limited computing resources and storage capacity, designing a secure and efficient authentication solution is quite challenging. Two key factors are as follows.

1) Lightweight: Most existing authentication protocols are based on elliptic curve or bilinear pairings. The protocols have high computational and communication overhead. For a large V2X communication network, lightweight solutions should be developed.

2) Mutual Authentication: Most existing authentication protocols only conduct unilateral authentication. For instance, the receiver of a message can confirm the identity of the sender, while the sender cannot confirm whether the receiver is a legitimate user. Mutual authentication can guarantee a secure communication.

E. Blockchain-Based Security Measures for Vehicular Networks

For the countermeasures elaborated in Section VI-B, there is another challenging factor, decentralization. Most existing authentication protocols need trusted third-party organizations to complete key distribution and authentication functionalities. The security of these authentication protocols heavily relies on third-party organizations. However, a centralized third-party organization is likely to be compromised. The concept of decentralization should be introduced.

Since V2X communication is conducted in VANET, V2X communication security can be tackled from another perspective, a vehicular network. As a powerful mathematical package, which is born with decentralization, blockchain has attracted much research attention [225]–[227].

Yang *et al.* [228] proposed a decentralized blockchain-based trust management system for vehicular networks. Messages

TABLE III
FEATURES AND ATTACK DEFENSE OF BLOCKCHAIN-BASED MEASURES

Scheme	Feature	Attack defense
[228]	trust management	message spoofing attack bad mouthing attack ballot stuffing attack data tempering
[229]	trust management	simple attack bad mouth attack zigzag attack
[230]	access control	identity masquerading
[231]	fairness & anonymity	free-riding attack double-claim attack repudiation attack forgeability attack DDoS attack & SPoF/C
[232]	miner selection	hash guessing attack transaction modification observe-act attack
[233]	key management	eavesdropping attack public key tampering attack DoS attack collusion attack
[234]	key negotiation	man-in-the-middle attack packet-dropping attack decryption failure attack
[235]	authentication	replay attack impersonation attack

received by a vehicle can be validated with neighboring vehicles by a Bayesian inference model. A block is constructed based on the validation results by RSUs.

Zhang *et al.* [229] developed an AI-enabled trust management system, which is similar to the proposal in [228]. The AI package used in the system is a deep learning algorithm.

Zheng *et al.* [230] proposed a blockchain-based secure computation offloading model for edge cloud offloading. To achieve consensus in vehicular networks, the authors developed a distributed hierarchical software-defined VANET (SDV) security architecture.

Li *et al.* [231] developed a fair and anonymous scheme for advertising in vehicular networks. The fairness is achieved using the Merkel hash tree and smart contracts. The anonymity is ensured with zero-knowledge proof techniques.

Kudva *et al.* [232] proposed a method called Proof of Driving (PoD). It is used to randomize the selection of honest miners for generating the blocks efficiently for blockchain-based VANET applications.

Ma *et al.* [233] developed a decentralized key management mechanism based on blockchain for VANET. The registration, update, and revocation of vehicle's public key are automatically conducted.

Chen *et al.* [234] proposed a traceable and authenticated key negotiation scheme based on blockchain. The scheme can be used for data sharing and electric transactions among vehicles.

Kaur *et al.* [235] developed a blockchain-based authentication mechanism for vehicular fog infrastructure. A cross-datacenter authentication and key-exchange scheme based on blockchain and elliptic curve cryptography (ECC) was elaborated.

For the above recent advances of blockchain-based security measures for vehicular networks, Table III shows their features and attacks, which could be defensed.

VII. DISCUSSION AND SOLUTION

A. Real-World Cases

Six representative cases of the four security dimensions elaborated previously are described as follows.

1) Sensor Security: In May 2016, a Tesla Model S with autopilot enabled in it crashed into a towed truck while turning left on a highway in Florida, USA, causing the driver's death [236]. The self-driving car was equipped with the Mobileye EyeQ3 vision system mounted in the middle of the windshield, a millimeter-wave radar under the front bumper, and 12 ultrasonic sensors around the body. The camera view on the Tesla car was blocked when the white truck turned, and at the same time, coupled with the interference of strong ambient light, the camera could not recognize the vehicles on the ground. The installation position of the millimeter-wave radar was too low, and the height of the chassis of the truck was higher than the detection distance of the millimeter-wave radar, which led to the failure of radar perception. In the case of an ultrasonic radar, since its measurement distance is generally short, it is impossible to detect longitudinal obstacles when driving at high speed. In general, this accident shows that the combination of the Mobileye vision system with the perception of the millimeter-wave radar is insufficient for solving the situation in the accident.

In June 2020, a Tesla car with autopilot enabled in it collided with a white truck [237]. Eight cameras and 12 millimeter-wave radars were installed around the car's body. The cameras were used for object recognition, whereas the radars were mainly used for measuring and following the speed of the vehicle on its front and its recognition rate for complex types of static objects was not high. In the sensor fusion process, only when the camera recognizes the vehicle in its front, can it be called the speed measurement information of the radar. This is because the camera recognizes obstacles based on the illumination and the physical colors of the surroundings. In this accident, the color of the vehicle in front of the car and the surrounding environment was similar. In addition, interference of strong ambient light led to an erroneous judgment of the camera. It is believed that there were no obstacles in front of the car. Another reason could be the limitation of the training data used in the camera vision algorithm. The deep learning model might not have been able to classify the top of the truck box, which led to the failure of its perception.

2) Operating System Security: In March 2018, in Arizona, USA, an unmanned Uber vehicle collided with a cyclist during a road test, causing the world's first unmanned driving accident in which a pedestrian died [238]. The unmanned vehicle was equipped with seven cameras, a 64-line LiDAR instrument, and multiple millimeter-wave radars. When the sensors detect pedestrians, the information is delivered to the central processing unit of the vehicle for processing in order to control the next move of the vehicle. According to the NTSB report, Uber found that the data collected by the camera, LiDAR, and radar on the unmanned vehicle were normal, and the LiDAR had detected the cyclist crossing the road 5.6 s before the accident. However, the classification of objects in the autonomous driving system was erroneous, and this led

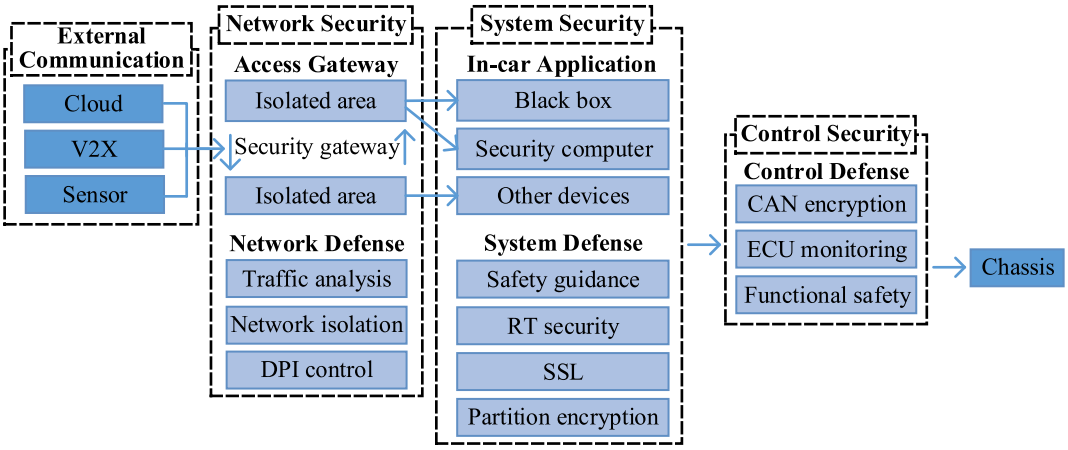

Fig. 6. Multilayered defense architecture.

to the failure of the software to correctly predict the victim's category and movement trajectory. The automated emergency braking system is generally required to turn on 1.3 s before the collision, but Uber disabled this function to prevent erratic driving [239].

3) Control System Security: In June 2015, two security experts, Charlie and Chris, used system vulnerabilities to remotely control a Chrysler Jeep car multimedia system to obtain permission for remotely sending commands to the CAN bus [240]. Without the user's knowledge, the driving speed of the car was reduced and ignition is turned off. Either the car engine suddenly braked or the brakes failed, causing 1.4 million vehicles to be returned to the factory for repair. This incident has exposed many security issues of the vehicle network, such as the use of the same cellular network for communicating with the device, lack of code signing, and no automatic update function, and these undoubtedly provide opportunities for hackers to attack vehicles. In addition, hackers can also use features such as the data flow entering the vehicle from the infrastructure to launch new attack channels against the vehicle.

4) V2X Communication Security: In November 2016, researchers from the Norwegian security service company Promon obtained the username and password of a Tesla APP account when they hacked into the user's mobile phone [241]. By logging into the Tesla IoVs service platform, they could locate, track, unlock, and start the vehicle at any time, eventually leading to the vehicle being stolen. In January 2018, a hacker attacked the data server of the car-sharing service provider GoGet, using the company's server to access the company's fleet and download user information, resulting in the leakage of a large amount of private data of car owners [242].

The main reason for the above two incidents is that the external network that the vehicle communicates with does not have a complete mechanism for encryption, authentication, and access control to prevent identity impersonation and information theft. Therefore, in the future design of the V2X communication network, in addition to reducing the network delay, it is necessary to strengthen the end-to-end encryption transmission, authentication, access control and abnormal traffic monitoring, and other security measures.

B. Conceptual Vehicle Information Security Framework

In order to ensure the safe operation of autonomous vehicles while driving, a real-time monitoring system for autonomous vehicles should be designed for monitoring the environmental status, the status of the vehicle itself, the status of the autonomous driving hardware and software, and the status of the driver. From the judgment of various state changes, the corresponding prompts, warnings, and triggers of the takeover strategy should be carried out in order to ensure that the process of automatic driving is always controllable, safe, and reliable. After a problem has been identified, it is necessary to provide an online diagnostic system to help users quickly determine the problem of the automatic driving system and provide feasible solutions to help users restore the system to a usable and safe state as soon as possible.

In the present work, we have constructed a vehicle information security solution based on the multilayer depth defense system. The main aim of the system is to "defend against external intrusions, prevent leakage of core applications and private data, and prevent threats of vehicle control." As shown in Fig. 6, the vehicle information security framework based on the multilayer depth defense system is mainly divided into six layers: 1) an external communication layer; 2) an access gateway layer; 3) a network defense layer; 4) an in-car application layer; 5) a system defense layer; and 6) a control defense layer.

1) External Communication Layer: The complete PKI system issues certificates for the devices participating in the automatic driving system and provides the required key and certificate management services. Secure communication is provided among the devices of the autonomous driving system and between the cloud and the car terminal to ensure confidentiality, integrity, authenticity, and tamper-proof communication data. The security upgrade kit ensures the safety and reliability of the OTA communication.

2) Access Gateway Layer: The dedicated vehicle security gateway isolates and controls access between the vehicle network and the Internet and vehicle subnetworks, and identifies instructions, detects and prevents abnormal network behavior and operation instructions of untrusted vehicles in order to ensure vehicle network security.

3) In-Car Application Layer: Based on the chip hardware security, from operating system guidance to running applications, a credibility measurement is performed throughout the entire process to prevent the operating system, core applications, and data from being tampered with. The privacy system provides protection for the core intellectual property (IP) and important business value data.

4) Network Defense Layer: The deep packet inspection (DPI) [243], [244] system is deployed on the IoVs platform to collect and analyze traffic and message content at key points in the vehicle network to detect abnormal network communication traffic and other behaviors and make audit records for subsequent security analysis.

5) System Defense Layer: By using security assessment, penetration testing, deployment of anti-distributed denial-of-service, WEB application firewall, and security log analysis tools on the cloud platform, the safe operation of the cloud platform is ensured. For mobile applications, the use of the memory obfuscation technology, patented virtual machine encryption technology, high strong protection shell technology, etc., to ensure that the application will not be used by hackers for vehicle attacks.

6) Control Security Layer: By encrypting CAN bus communication, it is ensured that the messages transmitted by the CAN bus of autonomous vehicles are not hijacked by malicious users. By monitoring the vehicle-mounted ECU, the monitoring module can determine whether a certain ECU is invaded by a malicious user, i.e., illegally obtaining the right to use the CAN bus. Functional safety ensures that the functions of the various components of the vehicle control system can be operated and run smoothly.

The vehicle extracts information from the cloud and the external environment using the external communication layer and transmits this information to the access gateway layer. This layer uses a dedicated vehicle security gateway that is divided into two isolation areas, which isolate the vehicle network from the Internet and vehicle subnetworks, and part of the information is transmitted to the black box and the security computer in the application layer of the car. For example, the data of various sensors are recorded, stored, and analyzed when the system requires the driver to control the car. Other equipment inside include the control of the accelerator and brake. The network defense layer uses the network isolation method to deploy the DPI system in the IoVs for analyzing abnormal network behavior and for using the dynamic defense system to monitor and block network attacks in time. Finally, the system defense layer uses RT management, secure sockets layer (SSL) certificates, partition encryption, and other operations to perform safety guidance of the vehicle. After passing through the system security layer, the vehicle needs to pass the executed instructions to control the security layer. The control security layer transmits control instructions to the actual components of the vehicle by encrypting CAN, ECU monitoring, and ensuring the functional safety of the control system.

For sensor security, we hold that actual sensor failure often shows its existence (e.g., obvious abnormal readings and sudden/dramatic changes of sensor data). On the contrary, the attacks against sensors are much sneakier. They tend to manipulate the sensor data and fool the high-level algorithms. Both actual sensor failure and attacks demand the autonomous driving system to operate in an error/intrusion-tolerant manner. We consider that sensors and sensor data can be covered in the "access gateway layer." This layer directly cooperates with the "external communication layer." There are some information, which cannot be included in a PKI system of the external communication layer, such as sunlight, rain, snow, fog, and shadows. Sensor data related to these phenomena can be cross-validated by multiple types of sensors with different data sources as described in Section III-F. In this case, both actual sensor failure and attacks may get compensated. Then, an autonomous driving system is expected to run in an error/intrusion-tolerant manner.

For operating system security, we hold that the major security drawback for the dominating operation system ROS2 in autonomous driving is it lacks protection for secure communication. We consider that the operating system security can be covered in the external communication layer. In this layer, the security of the operating system is first ensured from external communication and information, which flows in and out ROS2. Then, the OTA communication is secured by a correct use of the security upgrade kit. Moreover, the operating system security can be covered in the "in-car application layer." In this layer, the chip hardware security ensures that from operating system guidance to running applications, a credibility measurement is performed throughout the entire process to prevent the operating system, core applications, and data from being tampered with. The in-car application layer also possesses black box, security computer, and other devices. These components interact with each other based on the technologies (e.g., RT security and SSL) provided by the module "system defense" in Fig. 6. The above interactions are expected to achieve a secure operating environment for both the operating system and applications.

For control system security, we hold that the major drawbacks of existing protection methods for the control system are the significant computational cost and unsatisfactory real-time performance. We consider that control system security can be covered in the "control security layer." In this layer, the measures we adopted are effective choices, which are commonly accepted. These measures themselves do not show any improvement on computational cost or real-time performance. Nonetheless, the burdens on encryption and monitoring can be alleviated by partition encryption and RT security in the module system defense and the three features in the module "network defense" illustrated in Fig. 6.

For V2X communication security, we hold that the major drawback of existing countermeasures for the V2X communication is the requirement of authentication. As is known to all, communication protocols based on authentication schemes often possess high computational overhead, as well as extra communication overhead. Thus, lightweight solutions are needed. Besides high overhead, unilateral authentication is another flaw in most existing authentication schemes used for V2X communication in autonomous driving. Though distributed solutions (e.g., blockchain-based security measures described in Section VI-E) address the centralized problem, mutual authentication is still missing. Based on our investigation

and literature review, current authentication/encryption schemes are unable to possess completeness and robustness with a lightweight design. We consider that the V2X communication security can be covered in the external communication layer. In this layer, it is expected that the PKI system together with other supportive technologies is able to secure the V2X communication to some extent. Moreover, it can be covered in the "network defense layer." In this layer, traffic analysis conducted in the vehicular network is expected to detect abnormal communication and other behaviors, which might indicate a security issue.

VIII. Conclusion

Security is the primary requirement for autonomous driving. In this work, a retrospective and prospective study has been conducted in terms of four aspects: 1) sensor security; 2) operating system security; 3) control system security; and 4) V2X communication security. A detailed discussion of each attack path and the existing defense measures against these attack paths has been presented. The security problems of autonomous vehicles, caused by hackers intruding and tampering with data, belong to the category of information security, and thus, a conceptual framework has been proposed in this work to build an efficient vehicle information security. However, if an autonomous vehicle is to be mass produced, academia and industry still need to conduct additional research on the attack surface of autonomous driving modules. We hope that this article will attract attention in the computer and automobile circles.

Acknowledgment

The authors would like to thank the anonymous reviewers whose comments and suggestions greatly helped them improve the quality and presentation of this article. Cong Gao wants to thank his beloved mother, Miling Shen and family for their endless support and encouragement.

References

[1] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Jan. 2017.

[2] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The security of autonomous driving: Threats, defenses, and future directions," *Proc. IEEE*, vol. 108, no. 2, pp. 357–372, Feb. 2020.

[3] Q. Luo, Y. Cao, J. Liu, and A. Benslimane, "Localization and navigation in autonomous driving: Threats and countermeasures," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 38–45, Aug. 2019.

[4] S. Liu, L. Li, J. Tang, S. Wu, and J.-L. Gaudiot, *Creating Autonomous Vehicle Systems* (Synthesis Lectures on Computer Science), vol. 6. Williston, VT, USA: Morgan & Claypool, 2017, p. 186.

[5] "Apollo." Baidu Inc. [Online]. Available: https://github.com/ApolloAuto/apollo (Accessed: Jul. 5, 2021).

[6] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA) Workshop Open Source Softw.*, 2009, pp. 1–6.

[7] M. Howard. "Fending Off Future Attacks by Reducing Attack Surface." Feb. 2003. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/ms972812(v=msdn.10) (Accessed: Jul. 5, 2021).

[8] P. K. Manadhata and J. M. Wing, "An attack surface metric," *IEEE Trans. Softw. Eng.*, vol. 37, no. 3, pp. 371–386, May/Jun. 2011.

[9] P. Manadhata and J. M. Wing, "Measuring a system's attack surface," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Rep. CMU-CS-04-102,, Jan. 2004.

[10] P. K Manadhata, K. M. Tan, R. A. Maxion, and J. M. Wing, "An approach to measuring a system's attack surface," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Rep. CMU-CS-07-146, Aug. 2007.

[11] P. K. Manadhatam, "Game theoretic approaches to attack surface shifting," in *Moving Target Defense II*. New York, NY, USA: Springer, 2013, pp. 1–13.

[12] P. Manadhata, J. Wing, M. Flynn, and M. McQueen, "Measuring the attack surfaces of two FTP daemons," in *Proc. 2nd ACM Workshop Qual. Protection*, 2006, pp. 3–10.

[13] C. Theisen, N. Munaiah, M. Al-Zyoud, J. C. Carver, A. Meneely, and L. Williams, "Attack surface definitions: A systematic literature review," *Inf. Softw. Technol.*, vol. 104, pp. 94–103, Dec. 2018.

[14] S. Rizvi, R. J. Orr, A. Cox, P. Ashokkumar, and M. R. Rizvi, "Identifying the attack surface for IoT network," *Internet Things*, vol. 9, Mar. 2020, Art. no. 100162.

[15] C. Theisen, K. Herzig, B. Murphy, and L. Williams, "Risk-based attack surface approximation: How much data is enough?" in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. Pract. Track (ICSE-SEIP)*, 2017, pp. 273–282.

[16] N. Munaiah and A. Meneely, "Beyond the attack surface: Assessing security risk with random walks on call graphs," in *Proc. ACM Workshop Softw. Protection*, 2016, pp. 3–14.

[17] C. Theisen, K. Herzig, P. Morrison, B. Murphy, and L. Williams, "Approximating attack surfaces with stack traces," in *Proc. IEEE/ACM 37th Int. Conf. Softw. Eng.*, vol. 2, 2015, pp. 199–208.

[18] C. Maple, M. Bradbury, A. T. Le, and K. Ghirardello, "A connected and autonomous vehicle reference architecture for attack surface analysis," *Appl. Sci.*, vol. 9, no. 23, p. 5101, 2019.

[19] M. Salfer and C. Eckert, "Attack surface and vulnerability assessment of automotive electronic control units," in *Proc. 12th Int. Joint Conf. e-Bus. Telecommun. (ICETE)*, vol. 4, 2015, pp. 317–326.

[20] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. 20th USENIX Security Symp. (USENIX Security 11)*, vol. 4, 2011, pp. 447–462.

[21] Intel Transportation Solutions Division. "Research Summary of the Intel Automotive Security Research Workshops." 2016. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/automotive-security-research-workshops-summary.pdf (Accessed: Jul. 5, 2021).

[22] A. Chattopadhyay, K.-Y. Lam, and Y. Tavva, "Autonomous vehicle: Security by design," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7015–7029, Nov. 2021.

[23] D. Dominic, S. Chhawri, R. M. Eustice, D. Ma, and A. Weimerskirch, "Risk assessment for cooperative automated driving," in *Proc. 2nd ACM Workshop Cyber Phys. Syst. Security Privacy*, 2016, pp. 47–58.

[24] C. McCarthy, K. Harnett, and A. Carter, "Characterization of potential security threats in modern automobiles: A composite modeling approach," Nat. Highway Traffic Safety Administration United States, Washington, DC, USA, Rep. DOT HS 812 074, Oct. 2014.

[25] O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle, and B. Weyl, "Security requirements for automotive on-board networks," in *Proc. 9th Int. Conf. Intell. Transp. Syst. Telecommun. (ITST)*, 2009, pp. 641–646.

[26] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 546–556, Apr. 2015.

[27] K. Xiao, W. Shi, Z. Gao, C. Yao, and X. Qiu, "DAER: A resource preallocation algorithm of edge computing server by using blockchain in intelligent driving," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9291–9302, Oct. 2020.

[28] Y. Gu, L.-T. Hsu, and S. Kamijo, "Passive sensor integration for vehicle self-localization in urban traffic environment," *Sensors*, vol. 15, no. 12, pp. 30199–30220, 2015.

[29] G. Kar *et al.*, "Detection of on-road vehicles emanating GPS interference," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2014, pp. 621–632.

[30] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[31] Q. Zhu, L. Chen, Q. Li, M. Li, A. Nüchter, and J. Wang, "3D LiDAR point cloud based intersection recognition for autonomous driving," in *Proc. IEEE Intell. Veh. Symp.*, 2012, pp. 456–461.

[32] A. Singandhupe and H. La, "A review of SLAM techniques and security in autonomous driving," in *Proc. 3rd IEEE Int. Conf. Robot. Comput. (IRC)*, 2019, pp. 602–607.

[33] W. Xu, C. Yan, W. Jia, X. Ji, and J. Liu, "Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5015–5029, Dec. 2018.

[34] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: Approaches, lessons and challenges," *Philosoph. Trans. Roy. Soc. A, Math. Phys. Eng. Sci.*, vol. 368, no. 1928, pp. 4649–4672, 2010.

[35] M. Ilievski *et al.*, "Design space of behaviour planning for autonomous driving," 2019, *arXiv:1908.07931*.

[36] H. Cheng, *Autonomous Intelligent Vehicles: Theory, Algorithms, and Implementation*. London, U.K.: Springer, 2011.

[37] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.

[38] M. Zhang, N. Li, A. Girard, and I. Kolmanovsky, "A finite state machine based automated driving controller and its stochastic optimization," in *Proc. Dyn. Syst. Control Conf.*, 2017, pp. 1–10.

[39] S.-H. Bae, S.-H. Joo, J.-W. Pyo, J.-S. Yoon, K. Lee, and T.-Y. Kuc, "Finite state machine based vehicle system for autonomous driving in urban environments," in *Proc. 20th Int. Conf. Control Autom. Syst. (ICCAS)*, 2020, pp. 1181–1186.

[40] J. Liu and J. Liu, "Intelligent and connected vehicles: Current situation, future directions, and challenges," *IEEE Commun. Stand. Mag.*, vol. 2, no. 3, pp. 59–65, Sep. 2018.

[41] R. Marino, S. Scalzi, and M. Netto, "Nested PID steering control for lane keeping in autonomous vehicles," *Control Eng. Pract.*, vol. 19, no. 12, pp. 1459–1467, 2011.

[42] W. Farag and Z. Saleh, "Tuning of PID track followers for autonomous driving," in *Proc. Int. Conf. Innovation Intell. Informat. Comput. Technol. (3ICT)*, 2018, pp. 1–7.

[43] Q. Zhang *et al.*, "OpenVDAP: An open vehicular data analytics platform for CAVs," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 1310–1320.

[44] L. Liu *et al.*, "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6469–6486, Apr. 2021.

[45] Z. El-Rewini, K. Sadatsharan, N. Sugunaraj, D. F. Selvaraj, S. J. Plathottam, and P. Ranganathan, "Cybersecurity attacks in vehicular sensors," *IEEE Sensors J.*, vol. 20, no. 22, pp. 13752–13767, Nov. 2020.

[46] M. Hirz and B. Walzel, "Sensor and object recognition technologies for self-driving cars," *Comput.-Aided Design Appl.*, vol. 15, no. 4, pp. 501–508, 2018.

[47] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR," in *Proc. Black Hat Europe*, vol. 11, 2015, p. 995.

[48] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "Standard detectors aren't (currently) fooled by physical adversarial stop signs," 2017, *arXiv:1710.03337*.

[49] J. Zhang *et al.*, "Detecting and identifying optical signal attacks on autonomous driving systems," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1140–1153, Jan. 2021.

[50] Y. Cao *et al.*, "Invisible for both camera and LiDAR: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks," in *Proc. IEEE Symp. Security Privacy (SP)*, 2021, pp. 176–194.

[51] C. DiPalma, N. Wang, T. Sato, and Q. A. Chen, "Demo: Security of camera-based perception for autonomous driving under adversarial attack," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2021, p. 243.

[52] G. Rong *et al.*, "LGSVL simulator: A high fidelity simulator for autonomous driving," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, 2020, pp. 1–6.

[53] C. Kyrkou *et al.*, "Towards artificial-intelligence-based cybersecurity for robustifying automated driving systems against camera sensor attacks," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, 2020, pp. 476–481.

[54] C. Vitale *et al.*, "The CARAMEL project: A secure architecture for connected and autonomous vehicles," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2020, pp. 133–138.

[55] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn. (PMLR)*, 2017, pp. 1–16.

[56] M. M. Atia *et al.*, "A low-cost lane-determination system using GNSS/IMU fusion and HMM-based multistage map matching," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3027–3037, Nov. 2017.

[57] T. Li, H. Zhang, Z. Gao, Q. Chen, and X. Niu, "High-accuracy positioning in urban environments using single-frequency multi-GNSS RTK/MEMS-IMU integration," *Remote Sens.*, vol. 10, no. 2, pp. 205–216, 2018.

[58] R. T. Ioannides, T. Pany, and G. Gibbons, "Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques," *Proc. IEEE*, vol. 104, no. 6, pp. 1174–1194, Jun. 2016.

[59] V. L. L. Thing and J. Wu, "Autonomous vehicle security: A taxonomy of attacks and defences," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2016, pp. 164–170.

[60] J. Magiera and R. Katulski, "Detection and mitigation of GPS spoofing based on antenna array processing," *J. Appl. Res. Technol.*, vol. 13, no. 1, pp. 45–57, 2015.

[61] S. Han, L. Chen, W. Meng, and C. Li, "Improve the security of GNSS receivers through spoofing mitigation," *IEEE Access*, vol. 5, pp. 21057–21069, 2017.

[62] S. Dasgupta, M. Rahman, M. Islam, and M. Chowdhury, "Prediction-based GNSS spoofing attack detection for autonomous vehicles," 2020, *arXiv:2010.11722*.

[63] H. Schafer, E. Santana, A. Haden, and R. Biasini, "A commute in data: The comma2k19 dataset," 2018, *arXiv:1812.05752*.

[64] R. Mit, Y. Zangvil, and D. Katalan, "Analyzing Tesla's level 2 autonomous driving system under different GNSS spoofing scenarios and implementing connected services for authentication and reliability of GNSS data," in *Proc. 33rd Int. Tech. Meeting Satellite Division Inst. Navigation (ION GNSS+)*, 2020, pp. 621–646.

[65] S. Dasgupta, T. Ghosh, and M. Rahman, "A reinforcement learning approach for GNSS spoofing attack detection of autonomous vehicles," 2021, *arXiv:2108.08628*.

[66] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7699–7707.

[67] A. Broumandan and G. Lachapelle, "Spoofing detection using GNSS/INS/Odometer coupling for vehicular navigation," *Sensors*, vol. 18, no. 5, p. 1305, 2018.

[68] J. Song *et al.*, "Credible navigation algorithm for GNSS attack detection using auxiliary sensor system," *Appl. Sci.*, vol. 11, no. 14, p. 6321, 2021.

[69] S. H. Jeong *et al.*, "Low cost design of parallel parking assist system based on an ultrasonic sensor," *Int. J. Autom. Technol.*, vol. 11, no. 3, pp. 409–416, 2010.

[70] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicles," in *Proc. Def Con*, vol. 24, 2016, p. 109.

[71] B. S. Lim, S. L. Keoh, and V. L. L. Thing, "Autonomous vehicle ultrasonic sensor vulnerability and impact assessment," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, 2018, pp. 231–236.

[72] J. Lou, Q. Yan, Q. Hui, and H. Zeng, "SoundFence: Securing ultrasonic sensors in vehicles using physical-layer defense," 2021, *arXiv:2105.07574*.

[73] K. Ramasubramanian and K. Ramaiah, "Moving from legacy 24 GHz to state-of-the-art 77-GHz radar," *ATZelektronik Worldwide*, vol. 13, no. 3, pp. 46–49, 2018.

[74] P. Kapoor, A. Vora, and K.-D. Kang, "Detecting and mitigating spoofing attack against an automotive radar," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, 2018, pp. 1–6.

[75] C. Zhou, Q. Liu, and X. Chen, "Parameter estimation and suppression for DRFM-based interrupted sampling repeater jammer," *IET Radar Sonar Navigation*, vol. 12, no. 1, pp. 56–63, 2018.

[76] Z. Guan, Y. Chen, P. Lei, D. Li, and Y. Zhao, "Application of hash function on FMCW based millimeter-wave radar against DRFM jamming," *IEEE Access*, vol. 7, pp. 92285–92295, 2019.

[77] Z. Sun, S. Balakrishnan, L. Su, A. Bhuyan, P. Wang, and C. Qiao, "Who is in control? Practical physical layer attack and defense for mmWave based sensing in autonomous vehicles," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3199–3214, 2021.

[78] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against LiDARs for automotive applications," in *Proc. Int. Conf. Cryptogr. Hardw. Embedded Syst.*, 2017, pp. 445–467.

[79] Y. Cao *et al.*, "Adversarial sensor attack on LiDAR-based perception in autonomous driving," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2019, pp. 2267–2281.

[80] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust LiDAR-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *Proc. 29th USENIX Security Symp. (USENIX Security)*, 2020, pp. 877–894.

[81] R. Changalvala and H. Malik, "LiDAR data integrity verification for autonomous vehicle using 3D data hiding," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, 2019, pp. 1219–1225.

[82] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[83] K. Yang, T. Tsai, H. Yu, M. Panoff, T.-Y. Ho, and Y. Jin, "Robust roadside physical adversarial attack against deep learning in LiDAR perception modules," in *Proc. ACM Asia Conf. Comput. Commun. Security*, 2021, pp. 349–362.

[84] C. You, Z. Hau, and S. Demetriou, "Temporal consistency checks to detect LiDAR spoofing attacks on autonomous vehicle perception," 2021, *arXiv:2106.07833*.

[85] G. Sabaliauskaite, L. S. Liew, and J. Cui, "Integrating autonomous vehicle safety and security analysis using STPA method and the six-step model," *Int. J. Adv. Security*, vol. 11, nos. 1–2, pp. 160–169, 2018.

[86] M. Realpe, B. X. Vintimilla, and L. Vlacic, "A fault tolerant perception system for autonomous vehicles," in *Proc. 35th Chin. Control Conf. (CCC)*, 2016, pp. 6531–6536.

[87] N. Pous, D. Gingras, and D. Gruyer, "Intelligent vehicle embedded sensors fault detection and isolation using analytical redundancy and nonlinear transformations," *J. Control Sci. Eng.*, vol. 2017, pp. 1–10, 2017.

[88] Y.-S. Byun, B.-H. Kim, and R.-G. Jeong, "Sensor fault detection and signal restoration in intelligent vehicles," *Sensors*, vol. 19, no. 15, p. 3306, 2019.

[89] M. T. H. Anik, R. Saini, J.-L. Danger, S. Guilley, and N. Karimi, "Failure and attack detection by digital sensors," in *Proc. IEEE Eur. Test Symp. (ETS)*, 2020, pp. 1–2.

[90] A. Czarlinska and D. Kundur, "Attack vs. failure detection in event-driven wireless visual sensor networks," in *Proc. 9th Workshop Multimedia Security*, 2007, pp. 215–220.

[91] H. Utz, S. Sablatnog, S. Enderle, and G. Kraetzschmar, "Miro-middleware for mobile robot applications," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 493–497, Aug. 2002.

[92] J.-C. Baillie, "URBI: Towards a universal robotic low-level programming language," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 820–825.

[93] D. Calisi, A. Censi, L. Iocchi, and D. Nardi, "OpenRDK: A modular framework for robotic software development," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 1872–1877.

[94] A. Yousuf, C. C. Lehman, M. A. Mustafa, and M. M. Hayder, "Introducing kinematics with robot operating system (ROS)," in *Proc. Amer. Soc. Eng. Educ. Annu. Conf. Expo.*, 2015, pp. 26–1024.

[95] J. Będkowski, M. Pełka, K. Majek, T. Fitri, and J. Naruniec, "Open source robotic 3D mapping framework with ROS—Robot operating system, PCL—Point cloud library and cloud compare," in *Proc. Int. Conf. Elect. Eng. Informat. (ICEEI)*, 2015, pp. 1–18.

[96] Z. An, L. Hao, Y. Liu, and L. Dai, "Development of mobile robot SLAM based on ROS," *Int. J. Mech. Eng. Robot. Res.*, vol. 5, no. 1, pp. 47–51, 2016.

[97] S.-Y. Jeong *et al.*, "A study on ROS vulnerabilities and countermeasure," in *Proc. Companion ACM/IEEE Int. Conf. Human Robot Interact.*, 2017, pp. 147–148.

[98] J. McClean, C. Stull, C. Farrar, and D. Mascarenas, "A preliminary cyber-physical security assessment of the robot operating system (ROS)," in *Proc. SPIE Unmanned Syst. Technol. XV*, vol. 8741, 2013, Art. no. 874110.

[99] R. White, H. I. Christensen, and M. Quigley, "SROS: Securing ROS over the wire, in the graph, and through the kernel," 2016, *arXiv:1611.07060*.

[100] Q. Zhang, H. Zhong, J. Cui, L. Ren, and W. Shi, "AC4AV: A flexible and dynamic access control framework for connected and autonomous vehicles," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1946–1958, Feb. 2020.

[101] T. Kessler *et al.*, "Bridging the gap between open source software and vehicle hardware for autonomous driving," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2019, pp. 1612–1619.

[102] D. Heß *et al.*, "Contributions of the EU projects UnCoVerCPS and enable-S3 to highly automated driving in conflict situations," in *Proc. Annu. Conf. Amer. Assoc. Electrodiagnostic Technol.*, 2019, pp. 1–25.

[103] K. Xu, X. Xiao, J. Miao, and Q. Luo, "Data driven prediction architecture for autonomous driving and its application on Apollo platform," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2020, pp. 175–181.

[104] G. Pardo-Castellote, "OMG data-distribution service: Architectural overview," in *Proc. 23rd Int. Conf. Distrib. Comput. Syst. Workshops*, 2003, pp. 200–206.

[105] C. Eryigit and S. Uyar, "Integrating agents into data-centric naval combat management systems," in *Proc. 23rd Int. Symp. Comput. Inf. Sci.*, 2008, pp. 1–4.

[106] J. Kim, J. M. Smereka, C. Cheung, S. Nepal, and M. Grobler, "Security and performance considerations in ROS2: A balancing act," 2018, *arXiv:1809.09566*.

[107] "DDS Security." Object Management Group. Jul. 2018. [Online]. Available: https://www.omg.org/spec/DDS-SECURITY/1.1 (Accessed Jul. 5, 2021).

[108] Autoware Foundation. "Autoware." [Online]. Available: https://github.com/Autoware-AI/autoware.ai/ (Accessed Jul. 5, 2021).

[109] M. Reke *et al.*, "A self-driving car architecture in ROS2," in *Proc. Int. SAUPEC/RobMech/PRASA Conf.*, 2020, pp. 1–6.

[110] V. DiLuoffo, W. R. Michalson, and B. Sunar, "Robot operating system 2: The need for a holistic security approach to robotic architectures," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 3, pp. 1–15, 2018.

[111] Real-Time Innovations. "Software system integration with connext DDS professional." [Online]. Available: https://www.rti.com/products/connext-dds-professional (Accessed: Jul. 5, 2021).

[112] eProsima. "eProsima fast DDS." [Online]. Available: https://www.eprosima.com/index.php/products-all/eprosima-fast-dds (Accessed Jul. 5, 2021).

[113] ADLINK Technology Inc. "Data distribution service." [Online]. Available: https://www.adlinktech.com/en/data-distribution-service.aspx (Accessed Jul. 5, 2021).

[114] R. Morita and K. Matsubara, "Dynamic binding a proper DDS implementation for optimizing inter-node communication in ROS2," in *Proc. IEEE 24th Int. Conf. Embedded Real Time Comput. Syst. Appl. (RTCSA)*, 2018, pp. 246–247.

[115] Y. Maruyama, S. Kato, and T. Azumi, "Exploring the performance of ROS2," in *Proc. 13th Int. Conf. Embedded Softw.*, 2016, pp. 1–10.

[116] R. Herberth, S. Körper, T. Stiesch, F. Gauterin, and O. Bringmann, "Automated scheduling for optimal parallelization to reduce the duration of vehicle software updates," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2921–2933, Mar. 2019.

[117] J. Lawrence, "ROS2 prevalance and security," Rochester Inst. Technol., Rochester, NY, USA, Rep. CSEC 793, May 2020.

[118] S. Corrigan, "Introduction to the controller area network (CAN)," Texas Instrum., Dallas, TX, USA, Appl. Rep. SLOA101, Aug. 2002.

[119] J. Ning, J. Wang, J. Liu, and N. Kato, "Attacker identification and intrusion detection for in-vehicle networks," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1927–1930, Nov. 2019.

[120] B. Wang, S. Panigrahi, M. Narsude, and A. Mohanty, "Driver identification using vehicle telematics data," SAE Techn. Paper, Warrendale, PA, USA, Rep. 2017-01-1372, Jan. 2017.

[121] C. Young, J. Zambreno, H. Olufowobi, and G. Bloom, "Survey of automotive controller area network intrusion detection systems," *IEEE Des. Test*, vol. 36, no. 6, pp. 48–55, Dec. 2019.

[122] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," in *Proc. Black Hat USA*, vol. 91, 2015, pp. 1–9.

[123] A. Greenberg. "The Jeep hackers are back to prove car hacking can get much worse." Aug. 2016. [Online]. Available: https://www.wired.com/2016/08/jeep-hackers-return-high-speed-steering-acceleration-hacks/ (Accessed: Jul. 5, 2021).

[124] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Security Privacy*, 2010, pp. 447–462.

[125] S. Abbott-McCune and L. A. Shay, "Intrusion prevention system of automotive network CAN bus," in *Proc. IEEE Int. Carnahan Conf. Security Technol. (ICCST)*, 2016, pp. 1–8.

[126] M. A. Hannan, A. Hussain, and S. A. Samad, "System interface for an integrated intelligent safety system (ISS) for vehicle applications," *Sensors*, vol. 10, no. 2, pp. 1141–1153, 2010.

[127] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1697–1716, Aug. 2019.

[128] A. Groll and C. Ruland, "Secure and authentic communication on existing in-vehicle networks," in *Proc. IEEE Intell. Veh. Symp.*, 2009, pp. 1093–1097.

[129] L. Yu, J. Deng, R. R. Brooks, and S. B. Yun, "Automobile ECU design to avoid data tampering," in *Proc. 10th Annu. Cyber Inf. Security Res. Conf. (CISR)*, 2015, pp. 1–4.

[130] P.-S. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 395–399, Apr. 2014.

[131] Q. Wang and S. Sawhney, "VeCure: A practical security framework to protect the CAN bus of vehicles," in *Proc. Int. Conf. Internet Things (IoT)*, 2014, pp. 13–18.

[132] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle CAN," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 993–1006, Apr. 2015.

[133] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2016, pp. 63–68.

[134] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *Proc. World Congr. Ind. Control Syst. Security (WCICSS)*, 2015, pp. 45–49.

[135] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *Proc. 25th USENIX Security Symp. (USENIX Security)*, 2016, pp. 911–927.

[136] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2017, pp. 1577–1583.

[137] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, 2016, pp. 130–139.

[138] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS ONE*, vol. 11, no. 6, 2016, Art. no. e0155781.

[139] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown CAN bus networks," *Veh. Commun.*, vol. 9, pp. 43–52, Jul. 2017.

[140] Bosch Global, "CAN with flexible data-rate specification version 1.0," Robert Bosch GmbH, Gerlingen, Germany, Rep. 1.0, Apr. 2012.

[141] Bosch Global. [Online]. Available: https://www.bosch.com (Accessed: Jul. 5, 2021).

[142] F. Hartwich, "CAN with flexible data-rate," in *Proc. 13th Int. CAN Conf. (ICC)*, 2012, pp. 1–9.

[143] Bosch Global, "CAN specification version 2.0," Robert Bosch GmbH, Gerlingen, Germany, Rep. 2.0, Sep. 1991.

[144] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee, "A practical security architecture for in-vehicle CAN-FD," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2248–2261, Aug. 2016.

[145] G. Xie, L. T. Yang, W. Wu, K. Zeng, X. Xiao, and R. Li, "Security enhancement for real-time parallel in-vehicle applications by CAN FD message authentication," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5038–5049, Aug. 2021.

[146] G. Xie, R. Li, and S. Hu, "Security-aware obfuscated priority assignment for CAN FD messages in real-time parallel automotive applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 12, pp. 4413–4425, Dec. 2020.

[147] G. Xie, L. T. Yang, Y. Liu, H. Luo, X. Peng, and R. Li, "Security enhancement for real-time independent in-vehicle CAN-FD messages in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5244–5253, Jun. 2021.

[148] T. Yu and X. Wang, "Topology verification enabled intrusion detection for in-vehicle CAN-FD networks," *IEEE Commun. Lett.*, vol. 24, no. 1, pp. 227–230, Jan. 2020.

[149] Y. Xie, G. Zeng, R. Kurachi, H. Takada, and G. Xie, "Security/timing-aware design space exploration of CAN FD for automotive cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1094–1104, Feb. 2019.

[150] S. Fürst and M. Bechter, "AUTOSAR for connected and autonomous vehicles: The AUTOSAR adaptive platform," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw. Workshop (DSN-W)*, 2016, pp. 215–217.

[151] Y. Xiao, S. Shi, N. Zhang, W. Lou, and Y. T. Hou, "Session key distribution made practical for CAN and CAN-FD message authentication," in *Proc. 20th Annu. Comput. Security Appl. Conf.*, 2020, pp. 681–693.

[152] M. Agrawal, T. Huang, J. Zhou, and D. Chang, "CAN-FD-sec: Improving security of CAN-FD protocol," in *Security and Safety Interplay of Intelligent Software Systems*. Cham, Switzerland: Springer, 2018, pp. 77–93.

[153] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANET security challenges and solutions: A survey," *Veh. Commun.*, vol. 7, pp. 7–20, Jan. 2017.

[154] S. Park, B. Aslam, D. Turgut, and C. C. Zou, "Defense against sybil attack in the initial deployment stage of vehicular ad hoc network based on roadside unit support," *Security Commun. Netw.*, vol. 6, no. 4, pp. 523–538, 2013.

[155] J. Li, H. Lu, and M. Guizani, "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 938–948, Apr. 2015.

[156] Y. Yao *et al.*, "Multi-channel based sybil attack detection in vehicular ad hoc networks using RSSI," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 362–375, Feb. 2019.

[157] X. Feng, C.-Y. Li, D.-X. Chen, and J. Tang, "A method for defensing against multi-source sybil attacks in VANET," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 2, pp. 305–314, 2017.

[158] S.-H. Seo, J. Won, S. Sultana, and E. Bertino, "Effective key management in dynamic wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 371–383, Feb. 2015.

[159] T. Oulhaci, M. Omar, F. Harzine, and I. Harfi, "Secure and distributed certification system architecture for safety message authentication in VANET," *Telecommun. Syst.*, vol. 64, no. 4, pp. 679–694, 2017.

[160] J. T. Curran and A. Broumendan, "On the use of low-cost IMUs for GNSS spoofing detection in vehicular applications," in *Proc. Int. Tech. Symp. Navig. Timing (ITSNT)*, 2017, pp. 1–8.

[161] Q. Wang, Z. Lu, M. Gao, and G. Qu, "Edge computing based GPS spoofing detection methods," in *Proc. IEEE 23rd Int. Conf. Digit. Signal Process. (DSP)*, 2018, pp. 1–5.

[162] M.-C. Chuang and J.-F. Lee, "TEAM: Trust-extended authentication mechanism for vehicular ad hoc networks," *IEEE Syst. J.*, vol. 8, no. 3, pp. 749–758, Sep. 2014.

[163] A. Arsalan and R. A. Rehman, "Prevention of timing attack in software defined named data network with VANETs," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, 2018, pp. 247–252.

[164] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The Tesla broadcast authentication protocol," *RSA CryptoBytes*, vol. 5, no. 2, pp. 2–13, 2002.

[165] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient VANET authentication," *J. Commun. Netw.*, vol. 11, no. 6, pp. 574–588, 2009.

[166] P. Liu, B. Liu, Y. Sun, B. Zhao, and I. You, "Mitigating DoS attacks against pseudonymous authentication through puzzle-based co-authentication in 5G-VANET," *IEEE Access*, vol. 6, pp. 20795–20806, 2018.

[167] Y. Jie, M. Li, C. Guo, and L. Chen, "Dynamic defense strategy against DoS attacks over vehicular ad hoc networks based on port hopping," *IEEE Access*, vol. 6, pp. 51374–51383, 2018.

[168] Y. Gao, H. Wu, B. Song, Y. Jin, X. Luo, and X. Zeng, "A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 154560–154571, 2019.

[169] A. M. Malla and R. K. Sahu, "Security attacks with an effective solution for DoS attacks in VANET," *Int. J. Comput. Appl.*, vol. 66, no. 22, pp. 45–49, 2013.

[170] M. J. Faghihniya, S. M. Hosseini, and M. Tahmasebi, "Security upgrade against RREQ flooding attack by using balance index on vehicular ad hoc network," *Wireless Netw.*, vol. 23, no. 6, pp. 1863–1874, 2017.

[171] S. M. Safi, A. Movaghar, and M. Mohammadizadeh, "A novel approach for avoiding wormhole attacks in VANET," in *Proc. 2nd Int. Workshop Comput. Sci. Eng.*, vol. 2, 2009, pp. 160–165.

[172] S. Ali, P. Nand, and S. Tiwari, "Secure message broadcasting in VANET over wormhole attack by using cryptographic technique," in *Proc. Int. Conf. Comput. Commun. Autom. (ICCCA)*, 2017, pp. 520–523.

[173] A. Daeinabi and A. G. Rahbar, "Detection of malicious vehicles (DMV) through monitoring in vehicular ad-hoc networks," *Multimedia Tools Appl.*, vol. 66, no. 2, pp. 325–338, 2013.

[174] R. Baiad, O. Alhussein, H. Otrok, and S. Muhaidat, "Novel cross layer detection schemes to detect blackhole attack against QoS-OLSR protocol in VANET," *Veh. Commun.*, vol. 5, pp. 9–17, Jul. 2016.

[175] Z. A. Abdulkader, A. Abdullah, M. T. Abdullah, and Z. A. Zukarnain, "LI-AODV: Lifetime improving AODV routing for detecting and removing black-hole attack from VANET," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 1, pp. 1–15, 2017.

[176] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou, "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles," *Ad Hoc Netw.*, vol. 90, pp. 101823–101836, Jul. 2019.

[177] L. Mokdad, J. Ben-Othman, and A. T. Nguyen, "DJAVAN: Detecting jamming attacks in vehicle ad hoc networks," *Perform. Eval.*, vol. 87, pp. 47–59, May 2015.

[178] D. Karagiannis and A. Argyriou, "Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning," *Veh. Commun.*, vol. 13, pp. 56–63, Jul. 2018.

[179] L. He and W. T. Zhu, "Mitigating DoS attacks against signature-based authentication in VANETs," in *Proc. IEEE Int. Conf. Comput. Sci. Autom. Eng. (CSAE)*, vol. 3, 2012, pp. 261–265.

[180] A. Wasef, R. Lu, X. Lin, and X. Shen, "Complementing public key infrastructure to secure vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 17, no. 5, pp. 22–28, Oct. 2010.

[181] A. K. Malhi and S. Batra, "Genetic-based framework for prevention of masquerade and DDoS attacks in vehicular ad-hoc networks," *Security Commun. Netw.*, vol. 9, no. 15, pp. 2612–2626, 2016.

[182] W. Li and H. Song, "ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 960–969, Apr. 2016.

[183] M. A. Alazzawi, H. Lu, A. A. Yassin, and K. Chen, "Efficient conditional anonymity with message integrity and authentication in a vehicular ad-hoc network," *IEEE Access*, vol. 7, pp. 71424–71435, 2019.

[184] N.-W. Lo and H.-C. Tsai, "Illusion attack on VANET applications—A message plausibility problem," in *Proc. IEEE Globecom Workshops*, 2007, pp. 1–8.

[185] J. Zacharias and S. Fröschle, "Misbehavior detection system in VANETs using local traffic density," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, 2018, pp. 1–4.

[186] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient privacy-preserving authentication for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 907–919, Feb. 2014.

[187] P. Cencioni and R. Di Pietro, "VIPER: A vehicle-to-infrastructure communication privacy enforcement protocol," in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, 2007, pp. 1–6.

[188] C. Dai, X. Xiao, Y. Ding, L. Xiao, Y. Tang, and S. Zhou, "Learning based security for VANET with blockchain," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, 2018, pp. 210–215.

[189] G. Guette and B. Ducourthial, "On the sybil attack detection in VANET," in *Proc. IEEE Int. Conf. Mobile Ad Hoc Sensor Syst.*, 2007, pp. 1–6.

[190] T. Dimitriou, E. A. Alrashed, M. H. Karaata, and A. Hamdan, "Imposter detection for replication attacks in mobile sensor networks," *Comput. Netw.*, vol. 108, pp. 210–222, Oct. 2016.

[191] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 9, pp. 2467–2476, Sep. 2017.

[192] L. Bariah, D. Shehada, E. Salahat, and C. Y. Yeun, "Recent advances in VANET security: A survey," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, 2015, pp. 1–7.

[193] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, 2017, pp. 3–18.

[194] A. Rawat, S. Sharma, and R. Sushil, "VANET: Security attacks and its possible solutions," *J. Inf. Oper. Manag.*, vol. 3, no. 1, pp. 301–304, 2012.

[195] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[196] L. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[197] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETs): Status, results, and challenges," *Telecommun. Syst.*, vol. 50, no. 4, pp. 217–241, 2012.

[198] H. C. J. Lee and V. L. L. Thing, "Port hopping for resilient networks," in *Proc. IEEE 60th Veh. Technol. Conf. (VTC)*, vol. 5, 2004, pp. 3291–3295.

[199] K. Wang, J. Guo, and F. Li, "Singular linear space and its applications," *Finite Fields Their Appl.*, vol. 17, no. 5, pp. 395–406, 2011.

[200] I. A. Sumra, H. Bin Hasbullah, and J.-L. Bin AbManan, "Attacks on security goals (confidentiality, integrity, availability) in VANET: A survey," in *Proc. Vehicular Ad Hoc Netw. Smart Cities*, 2015, pp. 51–61.

[201] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Netw.*, vol. 61, pp. 33–50, Jun. 2017.

[202] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on VANET security challenges and possible cryptographic solutions," *Veh. Commun.*, vol. 1, no. 2, pp. 53–66, 2014.

[203] V. Bibhu, K. Roshan, K. B. Singh, and D. K. Singh, "Performance analysis of black hole attack in VANET," *Int. J. Comput. Netw. Inf. Security*, vol. 4, no. 11, pp. 47–54, 2012.

[204] D. Shukla, A. Vaibhav, S. Das, and P. Johri, "Security and attack analysis for vehicular ad hoc network—A survey," in *Proc. Int. Conf. Comput. Commun. Autom. (ICCCA)*, 2016, pp. 625–630.

[205] R. Rawat and D. Sharma, "Impact of jamming attack in vehicular ad hoc networks," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 4, pp. 457–461, 2015.

[206] A. Benslimane and H. Nguyen-Minh, "Jamming attack model and detection method for beacons under multichannel operation in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6475–6488, Jul. 2017.

[207] I. A. Sumra, H. Bin Hasbullah, I. Ahmad, and D. M. Alghazzawi, "Classification of attacks in vehicular ad hoc network (VANET)," *Information*, vol. 16, no. 5, pp. 2995–3004, 2013.

[208] J. De Fuentes, A. I. González-Tablas, and A. Ribagorda, "Overview of security issues in vehicular ad-hoc networks," in *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts*. Hershey, PA, USA: IGI Global, 2011, pp. 894–911.

[209] I. Ali, A. Hassan, and F. Li, "Authentication and privacy schemes for vehicular ad hoc networks (VANETs): A survey," *Veh. Commun.*, vol. 16, pp. 45–61, Apr. 2019.

[210] Z. Jianhong, X. Min, and L. Liying, "On the security of a secure batch verification with group testing for VANET," *Int. J. Netw. Security*, vol. 16, no. 5, pp. 351–358, 2014.

[211] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.

[212] M. S. Al-Kahtani, "Survey on security attacks in vehicular ad hoc networks (VANETs)," in *Proc. 6th Int. Conf. Signal Process. Commun. Syst.*, 2012, pp. 1–9.

[213] M. Y. Gadkari and N. B. Sambre, "VANET: Routing protocols, security issues and simulation tools," *IOSR J. Comput. Eng.*, vol. 3, no. 3, pp. 28–38, 2012.

[214] M. S. Sheikh and J. Liang, "A comprehensive survey on VANET security services in traffic management system," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–23, 2019.

[215] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, "Universal re-encryption for mixnets," in *Proc. Cryptograph. Track RSA Conf.*, 2004, pp. 163–178.

[216] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks," *Alexandria Eng. J.*, vol. 54, no. 4, pp. 1115–1126, 2015.

[217] "Network simulator version 2." [Online]. Available: http://www.isi.edu/nsnam/ (Accessed: Jul. 5, 2021).

[218] "Network simulator version 3." [Online]. Available: https://www.nsnam.org (Accessed: Jul. 5, 2021).

[219] "OMNeT++." [Online]. Available: https://www.omnetpp.org (Accessed: Jul. 5, 2021).

[220] "Glomosim." [Online]. Available: https://networksimulationtools.com/glomosim/. (Accessed: Jul. 5, 2021).

[221] Eclipse Foundation. "Simulation of urban mobility (SUMO)." [Online]. Available: https://www.eclipse.org/sumo/ (Accessed: Jul. 5, 2021).

[222] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: Generating realistic mobility patterns for VANETs," in *Proc. 3rd Int. Workshop Veh. Ad Hoc Netw.*, 2006, pp. 96–97.

[223] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: Realistic joint traffic and network simulator for VANETs," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 12, no. 1, pp. 31–33, 2008.

[224] *Matrix Laboratory (MATLAB)*, MathWorks Inc., Natick, MA, USA. https://www.mathworks.com/products/matlab/ (Accessed: Jul. 5, 2021).

[225] M. B. Mollah *et al.*, "Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021.

[226] R. Gupta, S. Tanwar, N. Kumar, and S. Tyagi, "Blockchain-based security attack resilience schemes for autonomous vehicles in Industry 4.0: A systematic review," *Comput. Elect. Eng.*, vol. 86, Sep. 2020, Art. no. 106717.
[227] X. Wang, C. Xu, Z. Zhou, S. Yang, and L. Sun, "A survey of blockchain-based cybersecurity for vehicular networks," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2020, pp. 740–745.
[228] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.
[229] C. Zhang, W. Li, Y. Luo, and Y. Hu, "AIT: An AI-enabled trust management system for vehicular networks using blockchain technology," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3157–3169, Mar. 2021.
[230] X. Zheng, M. Li, Y. Chen, J. Guo, M. Alam, and W. Hu, "Blockchain-based secure computation offloading in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4073–4087, Jul. 2021.
[231] M. Li, J. Weng, A. Yang, J.-N. Liu, and X. Lin, "Toward blockchain-based fair and anonymous Ad dissemination in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11248–11259, Nov. 2019.
[232] S. Kudva, S. Badsha, S. Sengupta, I. Khalil, and A. Zomaya, "Towards secure and practical consensus for blockchain based VANET," *Inf. Sci.*, vol. 545, pp. 170–187, Feb. 2021.
[233] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An efficient decentralized key management mechanism for VANET with blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5836–5849, Jun. 2020.
[234] Y. Chen, X. Hao, W. Ren, and Y. Ren, "Traceable and authenticated key negotiations via blockchain for vehicular communications," *Mobile Inf. Syst.*, vol. 2019, Dec. 2019, Art. no. 5627497.
[235] K. Kaur, S. Garg, G. Kaddoum, F. Gagnon, and S. H. Ahmed, "Blockchain-based lightweight authentication mechanism for vehicular fog infrastructure," in *Proc. IEEE Int. Conf. Commun. workshops (ICC Workshops)*, 2019, pp. 1–6.
[236] "Tesla model s driver killed in Williston Florida." 2016. [Online]. Available: https://www.thecarcrashdetective.com/joshua-brown-tesla-model-s-driver-killed-williston-fl/ (Accessed: Jul. 5, 2021).
[237] L. Manson. "Tesla autopilot makes model 3 crash into overturned truck." Jun. 2020. [Online]. Available: https://www.somagnews.com/tesla-autopilot-makes-model-3-crash-overturned-truck/ (Accessed: Jul. 5, 2021).
[238] N. A. Stanton, P. M Salmon, G. H. Walker, and M. Stanton, "Models and methods for collision analysis: A comparison study based on the Uber collision with a pedestrian," *Safety Sci.*, vol. 120, pp. 117–128, Dec. 2019.
[239] National Transpotation Safety Board. "Preliminary report highway: hwy18mh010." 2018. [Online]. Available: https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH010-prelim.pdf (Accessed: Jul. 5, 2021).
[240] A. Greenberg. "Hackers remotely kill a jeep on the highway—With me in it." Jul. 2015. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/ (Accessed: Jul. 5, 2021).
[241] D. Z. Morris. "Tesla-stealing hack is about much more than Tesla." Nov. 2016. [Online]. Available: https://fortune.com/2016/11/26/tesla-stealing-hack/ (Accessed: Jul. 5, 2021).
[242] D. Sadler. "Cyber pro charged with GoGet hacking." Feb. 2018. [Online]. Available: https://ia.acs.org.au/article/2018/cyber-pro-charged-with-goget-hacking.html (Accessed: Jul. 5, 2021).
[243] J. Sherry, C. Lan, R. Ada Popa, and S. Ratnasamy, "Blindbox: Deep packet inspection over encrypted traffic," in *Proc. ACM Conf. Special Interest Group Data Commun. (SIGCOMM)*, 2015, pp. 213–226.
[244] L. Deri and F. Fusco, "Using deep packet inspection in cyber traffic analysis," in *Proc. IEEE Int. Conf. Cyber Security Resilience (CSR)*, 2021, pp. 89–94.

Cong Gao received the Ph.D. degree in computer architecture from Xidian University, Xi'an, China, in 2015.

He is currently an Assistant Professor with the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an. His current research interests include data sensing and fusion, autonomous driving, and network security.

Geng Wang received the B.S. degree in software engineering from Shanxi University, Taiyuan, China, in 2018. He is currently pursuing the M.S. degree in software engineering with Xi'an University of Posts and Telecommunications, Xi'an, China.

His current research interests include autonomous driving and anomaly detection of sensor data.

Weisong Shi (Fellow, IEEE) received the Ph.D. degree in computer architecture from Chinese Academy of Sciences, Beijing, China, in 2000.

He is a Charles H. Gershenson Distinguished Faculty Fellow and a Full Professor of Computer Science with Wayne State University, Detroit, MI, USA. His current research interests include edge computing, computer systems for autonomous driving, mobile, and connected health.

Zhongmin Wang received the Ph.D. degree in mechanical engineering and automation from Beijing Institute of Technology, Beijing, China, in 2000.

He is currently a Professor with the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an, China. His current research interests include embedded intelligent perception, big data processing and application, and affective computing.

Yanping Chen received the Ph.D. degree in computer architecture from Xi'an Jiaotong University, Xi'an, China, in 2007.

She is currently a Professor with the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an. Her current research interests include service mining, service computing, and network management.