IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 August 2023, accepted 17 August 2023, date of publication 22 August 2023, date of current version 29 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307473

SURVEY

An Investigation of Cyber-Attacks and Security Mechanisms for Connected and Autonomous Vehicles

SANDEEP GUPTA[1], CARSTEN MAPLE[2], AND ROBERTO PASSERONE[1], (Member, IEEE)
[1]Department of Information Engineering and Computer Science (DISI), University of Trento, 38122 Trento, Italy
[2]Secure Cyber Systems Research Group (SCSRG), Warwick Manufacturing Group (WMG), The University of Warwick, CV4 7AL Coventry, U.K.

Corresponding author: Sandeep Gupta (sandeep.gupta@unitn.it)

This work was supported by the Fondazione Bruno Kessler (FBK) AI@TN Project.

ABSTRACT Connected and autonomous vehicles (CAVs) can fulfill the emerging demand for smart transportation on a global scale. Such innovations for transportation can bring manyfold benefits, from fully autonomous driving services to proactive vehicle monitoring and traffic management. However, given the complexity involved in the deployment of CAVs, zero-tolerance safety, and security measures must be incorporated to avert vehicle immobilization, road accidents, disclosure of sensitive data, or any potential threats. In this article, we conceive a reference architecture for a CAVs ecosystem to derive a common attack taxonomy for the investigation of existing and emerging cyber threats. Subsequently, we discuss security mechanisms for the CAVs ecosystem that can be useful for the safe and secure transportation of passengers from one destination to another based on comprehensive studies of academic literature and industry white papers. Our work can provide valuable insights to security engineers and system architects for investigating security problems using a top-to-bottom approach and can aid in envisioning robust security solutions to ensure seamless CAVs operations.

INDEX TERMS Connected and autonomous vehicles, edge computing, fog computing, cloud computing, cyber attacks, security mechanisms.

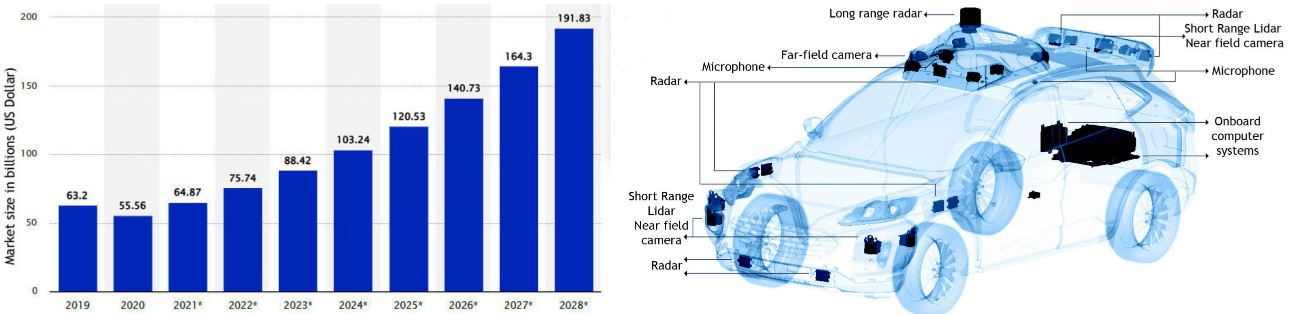
I. INTRODUCTION

Connected and autonomous vehicles (CAVs) will not only transform the existing automotive landscape but will also provide a highly connected infrastructure that is requisite for the emergence of smart transportation. With reference to Figure 1a, market studies report that the globally connected car market will be worth $191.83 billion by 2028 [1]. However, the growth of the CAV market can be derailed by increasing cyber security attacks targeting *Hardware*, *Software*, and *Network* systems [2], [3]. Thus, safety and security are paramount to ensure that CAVs can be operated cooperatively and free of hazards on the roads, and, at the same time, deliver a better user experience and confidence to drivers, passengers, and the public.

CAVs undertake numerous aspects of the dynamic driving tasks in all roadway and environmental conditions to automate the driving system [4]. As illustrated in Figure 1b, CAVs employ sensors, controllers, and onboard computers operated by sophisticated software and leverage wireless communication technologies to communicate with the surroundings [5], [6], [7]. However, security mechanisms implemented for smooth CAVs operations, secure vehicle-to-everything (V2X) communication, and seamless integration with support systems like Fog Computing (FC) and Cloud Computing (CC) require a thorough examination for the widespread acceptance of CAVs.

CAVs can not operate in isolation but indeed require supporting systems and services to be put in place, which can be achieved by an efficient and reliable ecosystem. A simplified three-tier-topology for a CAVs ecosystem is presented in Figure 4. The three tiers are 1) CAVs as Edge devices, 2) RSUs (Road-Side Units) as Fog, and 3) cloud

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.



(a) Connected cars global market size 2019-2028 [1]. The CAV market is estimated to reach 191.83 billion U.S. dollars by 2028.

(b) A typical connected and autonomous vehicle [8]

FIGURE 1. **Connected and autonomous vehicles.**

servers as the backbone infrastructure [9]. Both FC and CC can play a pivotal role in furnishing enhanced computing capabilities onboard and intermediate relay stations closer to CAVs that can be defined as Edge devices, for better operability [10]. It is estimated that a CAV will typically generate several terabytes of data in a single day of driving [11]. Consequently, powerful analytics programs will be required to extract actionable information from the immense amount of data being generated by CAVs. FC and CC can ensure that CAVs can process the data without any latency to take real-time decisions for reliable motion planning and controls.

Delays of a few milliseconds can be considered catastrophic for the safety of end-users. FC and CC can equip CAVs to adapt instantaneously to changing road conditions or environments, rather than relying on instructions or recommendations from distant cloud servers. Furthermore, FC and CC can expend high-speed V2X communications to achieve greater situational awareness of the events, potential threats, and imminent hazards making the future of CAVs both possible and practical. Seamless integration with other smart ecosystems will be essential for improving road safety, reducing congestion, upgrading transportation efficiency, enhancing mobility, increasing service reliability, optimizing energy consumption and environmental impacts, and supporting economic development [12].

In this article, we take a comprehensive approach to studying cyber attacks on CAVs that goes beyond considering them as a standalone entity or isolated system. Instead, we emphasize including the entire infrastructure that is crucial for secure and seamless CAVs operations. Thus, a reference architecture for the CAVs ecosystem is conceived that comprises three sub-architectures to instantiate each tier: *Cloud*, *Fog*, and *Edge*. Figure 2 clarifies our strategy and approach to carry out this work, where we first understand the high-level topology of a typical CAV ecosystem, then derive a reference architecture that we later use to classify the potential attacks and the available security mechanisms based on a unified attack taxonomy across the three tiers.

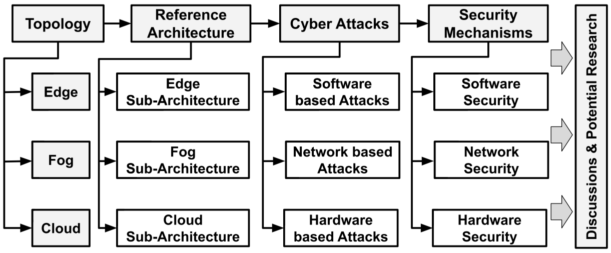


FIGURE 2. **Article's research strategy and approach.**

A. MOTIVATION AND SURVEY METHODOLOGY

In contrast to previous surveys on CAVs, we particularly focus on the security aspects of the entire CAVs ecosystem taking a holistic approach to investigate the security problems together with their potential solutions that can be highly useful for security engineers and system architects. Figure 3 shows the source-wise distribution of references used in this article. We perform a manual search using keywords *(e.g., CAV and security; CAV and denial of service)* to retrieve industry white papers and academic literature from leading journals and top conferences. A total of 528 publications are considered based on their title and are filtered down to 272 publications based on the abstract, however, 207 references are finally cited in the article according to their relevance. It can be observed that 57% of references are journal papers, 35% of references are conference papers, and 8% of references are white papers published since 2015 (*with a few exceptions of papers earlier than 2015*). Further, Table 2 discusses recent surveys that cover the topics related to security threats and vulnerabilities in CAVs and highlights their research topic and focus area.

B. CONTRIBUTIONS

The main contributions of this work are summarized as follows:

- We present a reference architecture of CAVs deriving a common attack taxonomy.

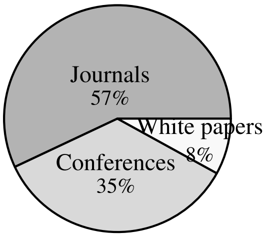


FIGURE 3. Source-wise references distribution (Total=207, journal=72, white papers=16).

- We describe the existing and emerging cyber-attacks in the CAVs.
- We describe potential security mechanisms that can be employed to secure a CAVs ecosystem.
- We finally, discuss the impact of cyber attacks on common security properties in the context of the CAVs ecosystem and outline the research trends and potential research directions.

C. ARTICLE ORGANIZATION

The rest of the article is organized as follows. Section II presents the related work that surveyed security threats and vulnerabilities in the context of CAVs and highlights how our survey differs from the previous ones. Section III describes CAVs three-tier topology and presents a reference architecture of CAVs. Section IV describes the existing and emerging cyber-attacks on CAVs. Section V provides security mechanisms for CAVs that can holistically address hardware, network, and software attacks. Section VI presents discussions and the potential research directions to evolve best practices for CAVs. Finally, Section VII concludes the article. Table 1 presents the acronyms and their description used in this article.

II. RELATED WORK

In recent years, several papers have been published that surveyed security threats and vulnerabilities in the domain of CAVs. Ju et al. [5] describe attack detection and resilience approaches for CAVs only from a vehicle dynamics and control perspective discussing three types of attacks, i.e., denial of service, reply, and false data injection attacks. The authors describe these attacks can be imposed on intra-vehicle networks, inter-vehicle networks, and perception sensors. Limbasiya et al. [6] present potential challenges, key security and privacy requirements, various capabilities of adversaries, and possible attacks in CAVs. Some of the attacks described are device tampering, unauthorized access, data forgery, and eavesdropping. Dibaei et al. [15] study Denial-of-Service (DoS), black-Hole, replay, Sybil, impersonation, malware, falsified information, and timing attacks in the vehicular networks context.

TABLE 1. Acronyms.

Acronyms	Description
AI	Artificial Intelligence
AES	Advanced Encryption System
BGP	Border Gateway Protocol
CAN	Controller Area Network
CAV	Connected and Autonomous Vehicle
CC	Cloud Computing
CNN	Convolutional Neural Network
COA	Ciphertext-Only Analysis
CPA	Chosen-Plaintext Analysis
CSI	Channel State Information
DDoS	Distributed Denial of Service
DoS	Denial-of-Service
DSRC	Dedicated Short-Range Communications
EC	Edge Computing
ECC	Elliptic Curve Cryptography
ECM	Engine Control Unit
ECUs	Electronic Control Units
ETC	Electronic Throttle Control
FC	Fog Computing
FPGA	Field Programmable Gate Array
FQDC	Fast Quartile Deviation Check
GPS	Global positioning system
HTM	Hierarchical Temporal Memory
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Systems
IoV	Internet of Vehicles
IPS	Intrusion Prevention System
ITS	Intelligent Transportation Systems
KPS	Known-Plaintext Analysis
LiDAR	Light Detection and Ranging
LIN	Local Interconnect Network
LSTM	Long Short-Term Memory
MAC	Media Access Control
MiTM	Man-in-The-Middle
ML	Machine Learning
MOST	Media Oriented Systems Transport
OBD	Onboard Diagnostic Port
OSI	Open Systems Interconnection
Radar	Radio Detection and Ranging
RSU	Road-Side Units
RSE	Roadside Equipment
RKP	Real-time Kinematic Positioning
RTL	Register Transfer Level
RSSI	Received Signal Strength Indicator
SCMS	Security Credential Management System
SDN	Software-Defined Network
SSL	Secure Socket Layer
TARA	Threat Analysis and Risk Assessment
TLS	Transport Layer Security
TSN	Time-Sensitive Networking
TSP	Telematics Service Providers
UDP	User Datagram Protocol
VANET	Vehicular Ad hoc Networks
V2V	Vehicle-to-Vehicle
V2R	Vehicle-to-RSU
V2X	Vehicle-to-Everything
ZKP	Zero-Knowledge Protocol

Aliwa et al. [7] describe in-vehicle serial bus protocols, e.g., CAN bus, FlexRay, Local Interconnect Network (LIN), and evaluate cryptographic and Intrusion Detection Systems (IDS) approaches used for protecting vehicular data. The authors state that the CAN protocol does not have security features and is vulnerable to attacks such as frame injection and denial of service. Considering the CAN Bus is the most widely used protocol to support critical functions such as power train, engine management, anti-brake system, and

TABLE 2. Recent CAVs surveys.

Reference	Year	Research topic covered	Focus area
Ju et al. [5]	2022	Describe attack detection and resilience approaches for CAVs only from vehicle dynamics and control perspective with a focus on denial of service, reply, and false data injection attacks.	Intra-vehicle communication network, perception sensors, and inter-vehicle communication network.
Limbasiya et al. [6]	2022	Present potential challenges, key security and privacy requirements, various capabilities of adversaries, and possible attacks in CAVs.	In-Vehicle Network and Controller Area Network.
Aliwa et al. [7]	2021	Describe in-vehicle serial bus protocols (e.g., CAN bus) and evaluate cryptographic and Intrusion Detection Systems (IDS) approaches used for protecting vehicular data.	In-vehicle serial bus protocols, evaluate cryptographic and Intrusion Detection Systems (IDS).
Liu et al. [9]	2021	Survey vehicular edge computing for the architecture, key enablers, advantages, and challenges.	Vehicular Edge Computing.
Pham et al. [13]	2021	Describe vulnerable CAV components and their exploitation by attackers. Attack models are discussed based on the targeted CAV components of attacks, access requirements, and attack motives.	CAV components such as sensors, cameras, and communication mechanisms.
Sun et al. [14]	2021	Discuss cyber-security risks and vulnerabilities in the environment of CAVs into in-vehicle network attacks, vehicle-to-everything network attacks, and related attacks.	In-vehicle network and vehicle-to-everything network.
Dibaei et al. [15]	2020	Discuss the security of intelligent vehicles classifying attacks, defenses, and vulnerabilities into four categories: cryptography, network security, software vulnerability detection, and malware detection.	Cryptography, network security, software vulnerability, and malware detection.
Sommer et al. [16]	2019	Automotive security attacks are classified showing how attacks can be represented at different levels.	Automotive security development process.
Le et al. [17]	2018	Analyzes security and privacy of automotive system architectures, applications, and application platforms.	In-vehicle systems, VANETs, and Internet-based applications.
Machardy et al. [18]	2018	Present the current state of research into V2X communication. It also discusses the relative pros and cons of dedicated short-range communications (DSRC).	V2X communication.
Wang et al. [19]	2018	Discuss the networking and communication technologies in autonomous driving.	Intra- and inter-vehicle network.

transmission, it is essential to secure the CAN Bus protocol. Liu et al. [9] provide an overview of Vehicular Edge Computing (VEC) including architecture, key enablers, advantages, and challenges. The authors specify that different vehicular users accessing the same physical edge servers can augment security and privacy without a strong protection mechanism. Consequently, vehicular networks in VEC can face new security and privacy challenges.

Pham et al. [13] describe vulnerable CAV components and their exploitation by attackers. The paper focuses on easy targets such as sensors, cameras, and communication mechanisms and attacks that can be executed at a distance from a CAV, i.e., from the roadside or other vehicles. The authors added defense strategies require further experiments to address new attack models that largely follow the remote attack pattern by targeting sensors, cameras, and communication mechanisms. Sun et al. [14] classify the cyber-security risks and vulnerabilities into in-vehicle network attacks, vehicle-to-everything network attacks, and other attacks according to the types of communication networks and attack objects. The paper highlights dynamic risks, a lightweight security model, trust levels, 5G cellular-based V2X security, and data collection and storage as some pressing challenges.

Sommer et al. [16] present a taxonomy to describe automotive security attacks that can be useful for threat analysis and risk assessment (TARA). Le et al. [17] investigate the main security and privacy challenges for the design of automotive applications and platforms. The paper identifies areas such as data collection, over-the-air updates, resilient in-vehicle networks, gateway firewalls, intrusion detection, and response that are required to be secure for the deployment of CAVs at a large scale. Machardy et al. [18] study V2X communication with a focus on the relative benefits and limitations of DSRC-based and mobile cellular network-based technologies. Wang et al. [19] investigate the networking and communication technologies enabling the perception and planning ability of CAVs based on the available sensors. Overall, it can be deduced from Table 2, the surveys on CAVs mainly cover the security of intra- and inter-vehicle networks and components such as sensors, cameras, and communication mechanisms. This article is an attempt to investigate existing and emerging cyber-attacks and their security solutions across the three tiers of the CAVs ecosystem based on a unified attack taxonomy.

III. CONNECTED AND AUTONOMOUS VEHICLES

CAVs can be described as data-powered vehicles that exploit connectivity and automated technologies for facilitating smart transportation on a global scale. Both Edge and Fog computing can be key technologies for processing the high-volume data that CAVs will generate every second of their operation to improve speed, safety, and reliability [20]. The edge and fog paradigm can be specified as a distributed and decentralized computing platform to process data at and near their origin [21]. In edge, computing occurs on the devices that interface sensors or gateways that are in close proximity to sensors, whereas, in fog, data processing occurs farther from the sensors. The motivation behind the processing of data closer to its origin is to address limitations, such as data processing latency, data loss due to poor connectivity, and network traffic congestion, observed in centralized computing platforms.

A. A THREE-TIER TOPOLOGY

Figure 4 presents a simplified tier-topology for a connected autonomous vehicles ecosystem in which the three tiers identify CAVs as Edge devices, RSUs as Fog, and cloud servers as the backbone infrastructure [9].

- *Edge* typically denote CAVs. CAVs are equipped with electronic control units (ECUs), an onboard diagnostic

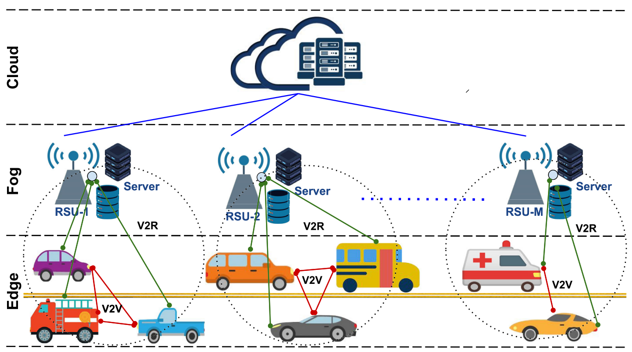


FIGURE 4. A simplified tier-topology of Connected and Autonomous Vehicles [9].

port (OBD), a controller area network (CAN), a global positioning system (GPS), light detection and ranging (LiDAR), radio detection and ranging (Radar), cameras (image sensors), etc., for real-time sensing of traffic and environmental conditions. Each CAV can exchange information with other CAVs in perceptible range using V2V (vehicle-to-vehicle) and with RSUs using V2R (Vehicle-to-RSU) communication. Additionally, CAVs have inbuilt computing and data storage capability to execute critical tasks locally. Thus, CAVs can be referred to as *prosumers*, i.e., they are not only a data consumer but also a data producer [22]. Tasks performed at this tier are 1) data sensing and collection, 2) data exchange, and 3) data processing to execute the assigned activities [23].

- *Fog* denotes Roadside Equipment (RSE) that includes RSUs, networking resources, and servers linking the edge devices and cloud systems [24]. RSUs facilitate real-time data analysis and intelligent raw data processing uploaded by the edge devices and further transfer the data to the cloud. They provide services like video streaming, traffic control, and path navigation for the smooth operation of CAVs. Fog enables computation offloading, task outsourcing, data caching, and software management operations [25]. Tasks performed at this tier are 1) local area information collection, filtering, aggregation, and cleansing, 2) analysis of local data with wide-area information, 3) real-time data processing, and 4) low-latency response to CAVs. Thus, the FC can be exploited to meet stringent performance requirements to automate driving.
- *Cloud* employs high-performance servers and storage devices to process and store the data uploaded by the edge devices. Cloud-based systems can be used for long-term storage and application-level data processing operations that are typically less time-sensitive [26]. Cloud services are provided for centralized management and control for taking optimal decisions.

B. REFERENCE ARCHITECTURE

Based on the simplified three-tier topology, a reference architecture of CAVs is construed. Figure 5 illustrates the reference the architecture of CAVs consists of three sub-architectures to instantiate each tier: *Edge*, *Fog*, and *Cloud*.

TABLE 3. CAVs: crucial systems and functionalities.

Systems	Functionalities
`Powertrain Controller` ⑪	Associated with an engine control unit (ECM), gear and traction control, and power steering for the management of vehicle dynamics [17].
`Chassis Safety Controller` ⑫	Associated with the anti-lock braking system, automatic stability control, adaptive suspension, adaptive cruise control, anti-slip regulation, airbags, etc., for driving dynamics, driving assistance, and active safety [17].
`Body Electronic Controller` ⑬	Associated with dashboard, wipers, lights, doors, windows, seats, mirrors, tire pressure system, climate control, etc., for managing non-dynamic vehicle components [17].
`Infotainment System` ⑭	Associated with entertainment systems (e.g., audio/video systems, music streaming, social media, etc.) and information systems (e.g., maps and navigation, smartphone and car status, etc.) [27].
`Navigation System` ⑮	Implements real-time navigation involves data sensing, collection, and processing of data from inertial and visual sensors [9]. Continually determining an optimal route for automated driving from the collected real-time traffic information.
`Sensor Systems` ㉒	On-board and in-car sensors, e.g., Radar, Lidar, Camera, Ultrasonic, Global Positioning, etc., collect the real-time surroundings data, such as road conditions, distance to other CAVs and objects, Global Navigation Satellite Systems (GNSS) positions to monitor the driving environment [19].
`Network Systems` ㉑	Manages both internal and external networks [28], e.g., Bluetooth, Wi-Fi, CAN, FlexRay, LIN, Media Oriented Systems Transport (MOST), eSIM, Ethernet gateway, communication devices (transceivers), etc.
`Power Management` ㉓	Responsible for the battery management, energy recovery, hybrid powertrain, actuators, etc. [28]

1) EDGE SUB-ARCHITECTURE

The *Edge sub-architecture* specifies crucial systems for real-time navigation and path planning together with situational and environmental awareness to enable safe and secure traffic maneuverability. Table 3 briefly describes the crucial systems, i.e., Powertrain Controller ⑪, Chassis Safety Controller ⑫, Body Electronic Controller ⑬, Infotainment System ⑭, Navigation System ⑮, Sensor Systems ㉒, Network Systems ㉑, Power Management ㉓, and their functionalities. Onboard computing systems ⑱ interact with each of these systems using an Internal Control Bus ⑯ to securely process various control units. Onboard computing systems also interact with the Data Analysis and Management module ⑳ that is driven by Artificial Intelligence (AI) ⑰ to implement all the features of CAVs. Further, onboard diagnostics systems ⑲ provide an external interface that allows plugging different maintenance and diagnostic devices into CAVs. Overall, a CAV interacts with other CAVs and other entities (e.g., road infrastructure, pedestrian, signal systems, etc.) providing V2V cooperative driving and maneuvering enhancements, cooperative collision warning systems, V2I/I2V route planning, V2I/I2V based variable speed limit/advisory, parking information and reservations to meet autonomous-driving objectives [29].

2) FOG SUB-ARCHITECTURE

The *Fog sub-architecture* specifies the systems augmenting the cloud systems to perform intermittent operations that can

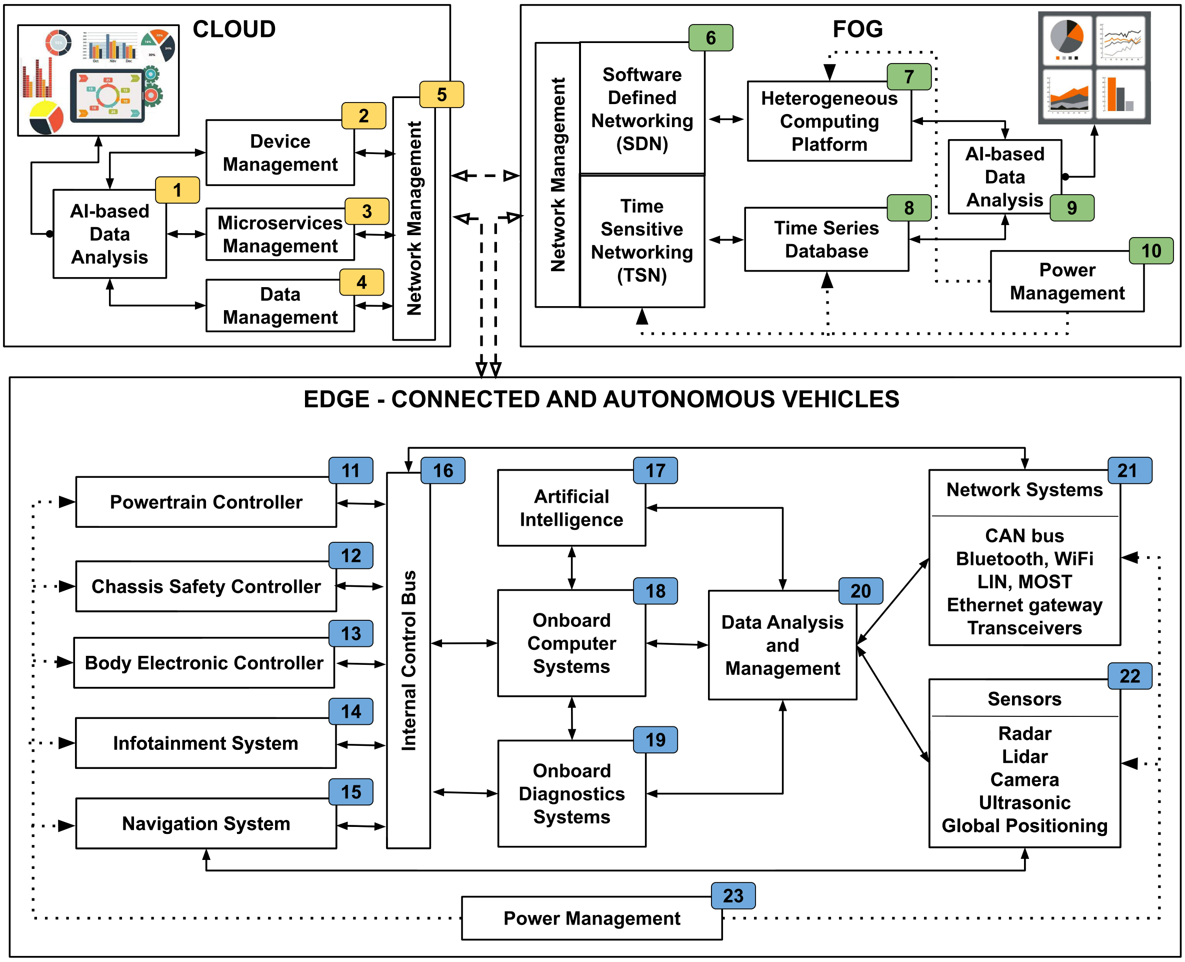


FIGURE 5. Connected and autonomous vehicles reference architecture.

reduce the network latency and enhance the computational performance of CAVs [10]. The key systems are network management 06, heterogeneous computing platform 07, time-series database 08, AI-based data analysis 09 and power management 10. Network management ensures ubiquitous network connections to enable a data exchange with minimal latency [30]. Software-defined network (SDN) separates the control plane from the data plane making the network programmable supporting access to millions of Edge virtualization functions. Time-sensitive networking (TSN) is responsible for resource reservation, clock synchronization, low-latency queue scheduling, path control, and configuration management.

A heterogeneous computing platform involves the integration of heterogeneous cores in a single system-on-chip to ensure low power consumption, high performance, portability, and cost-effectiveness. Time Series Database supports distributed storage, priority-based storage, and fragment-based query optimization offering efficient storage for time series data. Power management maintains power distribution units for uninterruptible power supplies in the Edge system. It also features remote control capabilities, advanced notifications, and environmental conditions monitoring. AI-based data analysis employs models to analyze internal systems to optimize their performance.

3) CLOUD SUB-ARCHITECTURE

In the *Cloud sub-architecture*, device-identity management 02, micro-services management 03, data management 04, and network management 05 are some vital systems for the smooth operation of CAVs. Device management is responsible for authenticating, authorizing, configuring, and monitoring the CAVs for persistent availability. Micro-services management secures critical services like high-density map production and deep-learning models for real-time situational awareness and traffic management. Data management facilitates a large amount of computing and storage requirements. Network management ensures a reliable and low-latency communication gateway along with end-to-end data and service synchronization support. AI-based data analysis 01 employs models to analyze time-series data generated from

CAVs to gain insights into patterns, events, tolerances, possible failures, or anomalies.

C. ATTACK TAXONOMY

A reference architecture can define families of technology components and their relationships. It can be useful in understanding and analyzing multi-tier systems dedicated to performing specified tasks to accomplish the given objectives within the given constraints [27]. Nonetheless, high-level abstraction of complex systems at the functional and communication levels can assist security specialists or subject matter experts in comprehensively assessing inherent vulnerabilities and potential threats for the generalization of possible attacks across all the tiers. Thus, by analyzing the reference architecture, we categorize different types of attacks within each tier into *software*, *network*, and *hardware* to derive a common attack taxonomy illustrated in Figure 6. Subsequently, we take a holistic approach to discuss existing and emerging cyber attacks and security mechanisms.

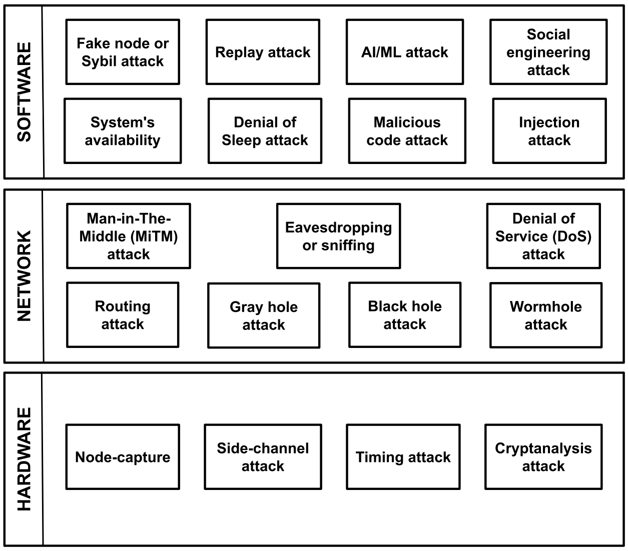


FIGURE 6. Attack taxonomy.

IV. EXISTING AND EMERGING CYBER ATTACKS ON CAVs

CAVs can be prone to both active or passive cyber attacks [31]. *Passive attacks*, e.g., eavesdropping, node destruction, node malfunctioning, node outage, traffic analysis, etc., are generally hidden or camouflaged. They harm or degrade the functioning of the system's vital components or collect information surreptitiously by tapping on the system communication links. On the other hand, *active attacks*, e.g., DoS, jamming, flooding, black hole, sinkhole, Sybil, wormhole, etc., can affect the functions and operations of the targeted system at once. Li et al. [32] described new threats in Telematics Service Providers (TSP) that can be an abuse of TSP services, insecure authentication, limited storage, and inadequate battery life. TSP integrates diverse communication systems in CAVs. The authors explain that vehicular botnets, authentication bypass, port intrusion, and malware implantation can be different attack methodologies for TSP.

In the rest of this section, we discuss the different types of attacks identified in our taxonomy. For each type, a table reports the possible impact of each threat and identifies the elements of the reference architecture of Figure 5 that can be most likely affected.

A. SOFTWARE ATTACKS

- *Fake node or Sybil attack*: Sybil attacks can be launched using various means, including software, network, or social engineering techniques. The adversary deploys fake or malicious nodes pushing bogus data to hinder the transmission of original information using multiple identities [33]. The presence of the Sybil node can affect the neighbor nodes by transmitting spurious data and can also compromise their privacy. In vehicular networks, Sybil attacks can be used to divert traffic in a certain direction [15]. This severe congestion at that location would force other CAVs to change their own routing to avoid congested areas. Techniques such as cryptographic methods, reputation-based systems, network monitoring, and anomaly detection algorithms can be employed to detect and mitigate the presence of fake identities and malicious behavior. Table 4 presents possible tier-wise impacts of a fake node or Sybil attack.

TABLE 4. Fake node or sybil attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
01	Device-identity management	✓		
05 06	Resource allocation and load balancing	✓	✓	
11-15 22	Trust and reputation			✓
04 09 20	Data integrity and security	✓	✓	✓
05 06 16 21	Communication protocols	✓	✓	✓
20-22	Collaborative computing			✓

- *Replay attack*: Replay attacks can be launched at the software, network, or hardware layers of a system, but they are primarily considered network or software attacks in the context of CAVs. The adversary manages to collect authentication information and re-transmits it illegitimately [34]. Eventually, the adversary deceives the receiver to perform unwanted actions. The common mechanisms to address replay attacks include encryption, authentication mechanisms, and timestamp verification. Table 5 presents possible tier-wise impacts of a replay attack.
- *AI/ML attack*: Attack on AI algorithms or machine learning (ML) models can be triggered in several ways: 1) manipulation of traffic signs to deceive traffic sign recognition of CAVs [16], 2) data falsification, e.g., GPS locations [35], or 3) false driving maneuver signals to mislead models for misclassifying an input [36]. Also, a poisoning attack can reduce the prediction accuracy of the learned model by injecting malicious samples

TABLE 5. Replay attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
20-22	Security protocols or authentication mechanisms			✓
06	RTU and gateways		✓	
03 07	Virtual machines, storage, microservices	✓	✓	
04 05 06 16 20	Data sources, network connectivity	✓	✓	✓

in the dataset that are used to train models [37], [38]. Table 6 presents possible tier-wise impacts of AI/ML attack. AI/ML attacks can be addressed using secure design, data validation, model validation, and continuous monitoring.

TABLE 6. AI/ML attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
04 08 20	Data integrity and decision-making processes	✓	✓	✓
01 09 17	AI models and algorithms	✓	✓	✓

- *Social engineering attack*: The adversary manipulates users to make security mistakes or give away sensitive information that can be used for breaching the authentication or access control mechanisms. Social engineering attacks rely on human error, such as baiting, scareware, pretexting, phishing, or spear-phishing that can be misused to attack the Edge computing paradigm [39]. These attacks can also be targeted for extracting users' sensitive data, thus, compromising their privacy. Table 7 presents possible tier-wise impacts of social engineering attacks. It is crucial to implement security awareness training, strong authentication mechanisms, regular security audits, and robust incident reporting and sharing dashboards.

TABLE 7. Social engineering attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
04 07 14 20	Authentication systems	✓	✓	✓
04 09 20	Data and privacy	✓	✓	✓
05 07 14 18	Third-Party services	✓	✓	✓

- *System's availability*: The adversary aims at degrading the data processing ability of the system, consequently, access to a system or its availability gets affected. Table 8 describes which can be the possible impact of the system's availability at each tier level. Redundancy measures, fault tolerance mechanisms, and backup strategies can be investigated to tackle the system's availability.
- *Denial of Sleep attack*: The adversary keeps nodes that are employed for periodically sensing data like temperature, humidity, vibration, etc., active to drain the power by denying them to go into the idle mode [40], [41]. Attacks like *Packet Flooding* or *Hello Flood* can be used for wasting network resources [42]. Table 9 presents possible tier-wise impacts of denial of sleep attack.

TABLE 8. System's availability.

Ref Arch 5	Impacts	Cloud	Fog	Edge
05 06 21	Routers, switches, and connectivity links	✓	✓	✓
03 07	Services and APIs	✓	✓	

TABLE 9. Denial of Sleep attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
22	Sensors			✓
06	Gateways		✓	
03	Virtual machines	✓		
10 23	Power management systems		✓	✓

- *Malicious code attack*: Infotainment systems connected to unsecured sources (e.g., web pages) can be compromised by malware. The adversary exploits a software program or script to create system vulnerabilities that can cause unwanted effects, security breaches, or system damage [43]. Table 10 presents possible tier-wise impacts of malicious code attacks that can be addressed using robust security measures, regular updates and patching, and secure coding practices.

TABLE 10. Malicious code attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
03 07 11-13 18	Firmware or operating system	✓	✓	✓
16 21	Communication protocols			✓
05 06	Network infrastructure	✓	✓	
02-04	Hypervisors, virtual machines, orchestration and management Systems	✓		

- *Injection attack*: The adversary injects a client-side script (XSS) like javascript in a trusted software application that can modify the application contents for deceiving or extracting the original information [44]. Table 11 presents possible tier-wise impacts of injection attack, which can be secured by implementing robust input validation, using parameterized queries, employing secure coding practices, and using formal verification methods.

TABLE 11. Injection attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
03 04 08 18 20	Services and databases	✓	✓	✓
03	Virtual machines, hypervisors, or container systems	✓		

B. NETWORK ATTACKS

- *Man-in-The-Middle (MiTM) attack*: The adversary intercepts and alters the genuine communication between the sender and the receiver without their knowledge, thus, manipulating both ends of information in real time. Conti et al. [45] studied MiTM attacks on different layers of open systems interconnection (OSI) models and types of cellular networks. MiTM attacks can be divided into four basic types, i.e., spoofing-based attacks, Secure Socket Layer (SSL) and Transport Layer Security (TLS) attacks, border gateway protocol (BGP) attacks, and false base station (FBS) attacks. Table 12 presents possible tier-wise impacts of MiTM attack, which needs robust encryption, strong authentication mechanisms, and secure communication protocols to tackle MiTM attack.

TABLE 12. MiTM attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
05 06 21	Communication Channels	✓	✓	✓
03 07	APIs, orchestrator for resource allocation and control traffic routing	✓	✓	

- *Eavesdropping or sniffing*: The adversary gathers vital information, such as the physical location of specific nodes, node identification or node configuration, message identities (IDs), timestamps, usernames, and passwords by tapping communication channels passively. Such intrusions can enable other attacks, e.g., fake node, replay attack, etc. Table 13 presents possible tier-wise impacts of eavesdropping or sniffing that encourage of deployment of strong data encryption and strict access control policies.

TABLE 13. Eavesdropping or sniffing.

Ref Arch 5	Impacts	Cloud	Fog	Edge
05 06 21	Communication channels and data integrity	✓	✓	✓

- *Denial of Service (DoS) attack*: The adversary floods the nodes with spurious requests to slow down or shut down an IoT ecosystem, thus, preventing users from accessing it [46]. DoS attacks employ techniques like flooding the target with UDP or ICMP packets to target various network devices, such as routers, switches, and firewalls. The packets are fake as they are spoofed and are full of random values [47]. *Distributed Denial of Service (DDoS) Attack* is another type of DoS attack that exploits multiple compromised nodes with unwanted traffic for breaking down an IoT ecosystem. Table 14 presents possible tier-wise impacts of DoS attack.

TABLE 14. DoS attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
05 06 21 22	Routers, RTUs, virtual machines, sensors	✓	✓	✓

- *Routing attack*: A routing loop is created by shortening or expanding the routing path through spoofing, redirecting, misdirecting, or even dropping packets [48]. As a result, the receiver node trusts the fake path instead of the valid one and routes some traffic toward the attacker. This attack can cause an end-to-end delay, affect uptime, and increase error messages. Table 15 presents possible tier-wise impacts of routing attacks that could lead to route hijacking, route poisoning, or distributed denial of service (DDoS) attacks, thus, security measures such as encryption, authentication mechanisms, and robust access controls are required.

TABLE 15. Routing attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
05 06	Network monitoring and management systems		✓	✓
03	Virtual private networks	✓		

- *Black- and Gray-hole attacks*: The adversary creates a fake node to redirect all the traffic to a proxy server or even discard it [49], [50]. The adversary achieves this by guaranteeing that the fake node has the shortest path, thus, accepting all the traffic. Gray hole attack is similar to the Blackhole attack but instead of dropping all of the packets, it drops only selected packet [51]. Table 16 presents possible tier-wise impacts of Black- and Gray-hole attacks.

TABLE 16. Black- and gray hole attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
03 05 06 21	Networks in CAVs and RTUs, and virtual machines	✓	✓	✓

- *Wormhole attack*: The adversary creates a tunnel by either controlling two different nodes in the network or adding new fake nodes to the network [52]. Consequently, data can be collected from one node and replayed using the other node to misguide network traffic. Table 17 presents possible tier-wise impacts of Wormhole attack.

TABLE 17. Wormhole attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
03 04	VMs and data centers	✓		
03 08 20	Data integrity and confidentiality	✓	✓	✓

C. HARDWARE ATTACKS

- *Node-capture*: The adversary physically controls key nodes, e.g., a gateway or a base node, and then reprograms and redeploys them to carry out various attacks [53]. It involves physically accessing and potentially tampering with the hardware of the targeted device. Consequently, information including communication between the sender and receiver gets leaked. A distributed computing paradigm like Edge computing can be easily prone to node capture due to the unattended deployment across a large terrain. Secure authentication, access controls, network segmentation, and intrusion detection systems can be leveraged to minimize the impacts of node-capture as shown in Table 18.
- *Side-channel attack*: The adversary applies forensic techniques to extract information such as execution

TABLE 18. Node-capture.

Ref Arch 5	Impacts	Cloud	Fog	Edge
08 20	Unauthorized access to data and services		✓	✓
07 11-15 18	Service disruptions and unauthorized actions		✓	✓
03	Virtual machines	✓		

time, power consumption, power dissipation, and electromagnetic interference [54]. These are noninvasive hardware-based attacks that target the physical implementation of the embedded hardware to obtain secret keys used for the encryption processes. Countermeasures such as cryptographic protections, isolation techniques, proper system configurations, and secure implementation practices can be employed to address side-channel attacks shown in Table 19.

TABLE 19. Side-channel attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
11-13 18 19	System on chip, embedded systems	✓		
06 21	Communication interfaces and links		✓	✓
03 05	Virtualization infrastructure and communication Channels	✓		

- *Timing attack*: Timing attacks are a type of side-channel attack that can be performed both through software and hardware. The adversary tries to discover vulnerabilities in the security mechanisms by observing a node's response time to various queries, input, or cryptographic algorithms [55]. The adversary targets nodes with weak computing capabilities to implement timing attacks. Robust cryptographic algorithms, secure coding practices, and proper timing mechanisms can be implemented to minimize the impact of timing attacks presented in Table 20.

TABLE 20. Timing attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
07 11-13 18	Cryptographic processes		✓	✓
03	Virtual machines and hypervisors	✓		
04 05 06 08	Storage systems, networking devices, or load balancing	✓	✓	

- *Cryptanalysis attack*: Cryptanalysis can exploit statistical analysis, mathematical analysis, and brute force. The adversary examines the cipher text to exploit the weaknesses or vulnerabilities in the cryptography algorithm to break down into the systems [56]. Cryptanalysis attacks include Known-Plaintext Analysis (KPA), Chosen-Plaintext Analysis (CPA), Ciphertext-Only Analysis (COA), or Man-in-the-Middle (MITM) Attack [57]. Regular updates, strong cryptographic algorithms, and secure key management practices are essential for addressing the impacts of Cryptanalysis attack Table 21.

TABLE 21. Cryptanalysis attack.

Ref Arch 5	Impacts	Cloud	Fog	Edge
04 05 06 18	Encryption Algorithms and random number generators	✓	✓	✓
02 07 20	Key management systems	✓	✓	✓
05 06 21	Cryptographic protocols, hash functions and libraries	✓	✓	✓
07 18 22	Hardware security modules		✓	✓

V. SECURITY MECHANISMS FOR CAVs

We study the recent state-of-the-art to investigate security mechanisms for CAVs in the context of the security of the Edge, Fog/Cloudlets, or Cloud computing paradigm. After that, we present a detailed analysis of software security solutions, encryption mechanisms security, network security, and physical nodes/devices security that can be applied in addressing cyberattacks discussed in Section IV, thus, securing a CAVs ecosystem.

A. SOFTWARE SECURITY

A large number of software/firmware/embedded applications have been employed to operate CAVs. Le et al. [17] analyze security and privacy issues related to in-vehicle systems, Vehicular Ad hoc Networks (VANETs), and Internet-based applications covering automotive system architectures, applications, and application platforms.

1) DEFENSES AGAINST FAKE NODE OR SYBIL ATTACK

Sybil attack is essentially an impersonation attack that can have three possible orthogonal dimensions, i.e., direct vs. indirect communication, fabricated vs. stolen identities, simultaneous vs. non-simultaneous [58]. With reference to Table 4, fake node or Sybil attack can impact 01 04 05 06 09 11-16 20-22 shown in Figure 5 and Table 22 presents a synopsis of defenses against fake node and Sybil attack.

TABLE 22. Defenses against fake node and sybil attack.

Mechanism	Reference
Symmetric cryptography.	Vasudeva & Sood [59]
Strong authentication methods.	Pham et al. [13]
Entropy theory for predicting traffic distribution.	Li et al. [60]
Location-hidden authorized message generation scheme.	Chang et al. [61]
Channel state information	Wang et al. [62]
Position estimation, distribution verification, or similarity comparison.	Yao et al. [63]
redit mechanisms to design routing algorithms.	Zhang and Li [64]
Social graph-based or behavior classification-based detection.	Zhang et al.[65]
Cryptographic digital signature certificate.	Reddy et al. [66]
Semi-supervised learning framework.	Gong et al. [67]
ML-based classification schemes.	Yang et al. [68]
Deep learning-based anomaly detection.	James [69]

Vasudeva and Sood [59] discuss potential mechanisms to prevent fake node and Sybil attacks that include symmetric cryptography using a central authority, random key predistribution (key pool, single-space pairwise, and multi-space pairwise), radio resource testing, received signal strength indicator, time difference of arrival, neighborhood data, passive ad hoc Sybil identity detection, passive ad hoc Sybil identity with group detection, and energy trust-based system. Pham et al. [13] describe strong authentication methods that can address impersonation or Sybil attacks preventing adversaries to send falsified information over the V2X communication channels to disrupt CAVs operation and traffic flow.

Li et al. [60] propose a Real-Time Edge Detection Scheme for Sybil DDoS that uses the entropy theory to quantify the traffic distribution and further design an algorithm named Fast Quartile Deviation Check (FQDC) to recognize and locate the attack on the Internet of Vehicles (IoV). *Footprint* is a Sybil attack detection mechanism that uses the trajectories of vehicles for identification without affecting the anonymity and location privacy of vehicles [61]. The authors explain that a location-hidden authorized message generation scheme in the *Footprint* can serve two purposes: 1) an RSU will be anonymous at the time of signing a message, i.e., the RSU location information is concealed from the final authorized message, and 2) authorized messages are temporarily linkable, i.e., two authorized messages issued signed by the same RSU remain valid only if they are issued within the same period.

Wang et al. [62] propose a Sybil attack detection method based on Channel State Information (CSI) to determine if the static devices are Sybil attackers. The method combines a self-adaptive multiple signal classification algorithm with the Received Signal Strength Indicator (RSSI). The authors also design a tracing scheme to cluster the channel characteristics of devices and detect dynamic attackers that change their channel characteristics in an error area. Similarly, Yao et al. [63] propose RSSI-based Sybil node detection that applies position estimation, distribution verification, or similarity comparison to identify Sybil nodes. Zhang and Li [64] study routing algorithms based on credit mechanisms to detect Sybil attacks. Zhang et al. [65] propose social graph-based Sybil detection (SGSD), behavior classification-based Sybil detection (BCSD), and mobile Sybil detection to defend against Sybil attacks.

Reddy et al. [66] propose a cryptographic digital signature certificate method to establish trust between participating entities. The method assigns a set of Public/Private Key pairs to each CAV that can be used by a CAV to authenticate itself to receivers by digitally signing the messages. Gong et al. [67] propose a semi-supervised learning framework (SybilBelief) to perform both Sybil classification and Sybil ranking. The authors evaluate the impact of factors like parameter settings in SybilBelief, the number of labels, and label noise on the performance of SybilBelief using synthetic networks.

Yang et al. [68] propose a classification scheme that incorporates the naive Bayes, decision tree, and support vector machine to detect Sybil attackers according to their mobility behaviors. The authors develop a community-based collision detection scheme based on fine-grained vehicle trajectories to alleviate the collusion among multiple Sybil attackers. James et al. [69] proposes a deep generative model for Sybil attack identification using Bayesian deep learning. The model exploits time-series features to embed trajectories in a latent distribution space, which serves as a basis for identifying ones generated by Sybil attacks.

2) DEFENSES AGAINST REPLAY ATTACK

Replay attacks can be used to manipulate the vehicle locations by re-transmitting the previous messages [70]. The adversary can target LiDAR [71], CAN bus [13], keyless entry systems [72], authorization and key agreement protocols [14], or communication between CAVs and RSU [15] in CAVs that can further lead to unauthorized data transmission, Eavesdropping, or Sniffing. With reference to Table 5, replay attack can impact 03-05 06 07 16 20-22 shown in Figure 5 and Table 23 presents a synopsis of defenses against replay attacks.

TABLE 23. Defenses against replay attack.

Mechanism	Reference
Prevent spoofing and jamming.	El et al. [73]
Distributed firewall for each sensor and communication module.	Kim et al. [74]
Formal verification of network protocols.	Shi et al. [75]
Timestamp mechanisms.	Dai et al. [76], Greene et al. [77]
Filtering and watermarking techniques.	Marquis et al. [78]
Real-time anomaly detection.	Wang et al. [79]
Access over-privileged detection.	Hong et al. [80]
ML-based anomaly detection based on contextual information.	Ashraf et al. [81]

Defense mechanisms to prevent spoofing and jamming can minimize the occurrence of a replay or relay attack [73]. Kim et al. [74] describes how to use a hybrid security system and a distributed firewall for each sensor and communication module, e.g., GPS, Bluetooth, and Wi-Fi for the in-vehicle network. Shi et al. [75] devise a formal method to verify a protocol's ability to resist replay attacks. The method uses formal languages to establish the attack models for the protocol and to analyze the network protocol communication process. The method traverses all the states of the protocol to determine the design flaws in the protocol that can help protocol designers identify the inherent weaknesses.

Dai et al. [76] propose a timestamp mechanism to prevent command replay attacks. The authors suggest that the client and server can negotiate a valid time interval in advance by setting a block generation time during which the messages are considered valid. Any identical messages in the same block can be discarded. Similarly, Greene et al. [77] propose a timestamp-based defense mechanism integrated with rolling code to mitigate replay attacks in existing remote keyless entry (RKE) systems.

Marquis et al. [78] propose a resilient estimator that combines Kalman filtering and watermarking to mitigate replay and spoofing attacks on sensors in cyber-physical systems (CPS). The authors describe how the method can estimate the correct state of the system by proactively adjusting the variance of potentially compromised sensors. Wang et al. [79]

propose a distributed real-time anomaly detection system based on a hierarchical temporal memory (HTM) learning algorithm. The HTM network learns the time-based data sequence in a continuous online manner for prediction, classification, and anomaly detection.

Hong et al. [80] present an AVGuardian tool that can detect over-privileged instances in autonomous vehicle software. The authors construct three different types of attacks by exploiting vulnerabilities resulting from over-privileged problems in-vehicle systems to demonstrate the severity of over-privileged access. The tool can generate the corresponding access control policies at the message field granularity to perform online policy violation detection and prevention. Ashraf et al. [81] describe a statistical feature extraction technique to acquire contextual features from network traffic that can be exploited by IDS for IoVs. The authors also propose a long short-term memory (LSTM) autoencoder-based scheme to design an IDS.

3) DEFENSES AGAINST AI/ML ATTACK

Any attacks to mislead or override the decisions taken by AI/ML systems can be devastating to the CAVs ecosystem. Sharma et al. [35] analyze VeReMi, a labeled simulated dataset providing a wide range of traffic behavior and attacker implementations, and measure the performance of different machine learning and deep learning models. Rauber et al. [82] design Foolbox, which is a Python-based library to benchmark the robustness of ML models against adversarial perturbations. The authors claim that the Foolbox can compare the robustness of ML models implemented using different frameworks.

Gao et al. [83] propose queue length estimation-based defense against data poisoning attacks that can be a serious threat to intelligent transportation systems. The authors analyze the characteristics of the single-point attack and increase the number of attack points to analyze the system vulnerabilities. Wang et al. [84] investigate the security properties of ML algorithms under adversarial settings. With reference to Table 6, AI/ML attack can impact 01 04 08 09 17 20 shown in Figure 5 and Table 24 presents a synopsis of attacks on different ML processes and the countermeasures.

TABLE 24. Synopsis of attacks on ML processes and the countermeasures.

ML Process	Attacks	Countermeasures
Training/Testing data alteration, Retraining, Learning algorithm, Feature extraction	Causative or poisoning attack [85] (e.g., Label-flipping, Clean Label, Backdoor, Trojans), Mimicry Attack, Data perturbation, Gradient descent attack, Polymorphic, Metamorphic [85]	Data sanitization techniques [86], [87], [88], Adversarial sample thwarting (e.g., data transformation, noise filtering, mapping to normal samples) [89], Generalization enhancement (e.g., bagging, random subspace method, antidote, etc.) [84], Training data filtering (e.g., input manipulation detection, gradient shaping) [38], Robust learning (e.g., model robustifying and verification) [38], Feature obfuscation [90], Active learning systems [90], Min-max optimization [91], [92].
Altering classifier decision, Tempering detection model	Evasion attack [85] (e.g., Fast Gradient Sign Method, Limited-memory Broyden–Fletcher–Goldfarb–Shanno, Universal Attack Approach, Universal Perturbations for Steering to Exact Targets, Antagonistic Network for Generating Rogue, DeepFool, Jacobian-based Saliency Map), Model stealing.	Gradient-based approach [93], Enhance generalization capability using adversarial feature selection method [94]. Adversarial training or distillation algorithm to improve the robustness of the machine learning model [37]. Remove redundant/irrelevant features used in ML models [95]. Data transformations (e.g., linear dimensionality reduction techniques like Principal Component Analysis) [96]

4) DEFENSES AGAINST SOCIAL ENGINEERING ATTACK

Social engineering attacks, i.e., manipulation and exploitation of people, can trick administrators, operators, and end-users to extract sensitive information [97], [98]. Social engineering attacks can involve the creation of fake RTU-CAV messages, phishing, or GPS map poisoning. Social engineering attacks can be addressed by providing security training and awareness against such types of attacks [99]. The training and awareness programs can develop stakeholders skills to identify, tackle, and report any social engineering malicious attempts. With reference to Table 7, social engineering attack can impact 04 05 07 09 14 18 20 shown in Figure 5 and Table 25 presents a synopsis of defenses against social engineering attacks.

TABLE 25. Defenses against social engineering attacks.

Mechanism	Reference
Human-as-a-security-sensor.	Heartfield and Loukas [97]
Human behavioral analysis.	Fan et al. [100]

Heartfield and Loukas [97] develop a human-as-a-security-sensor (HaaSS) framework that leverages the ability of human users to act as sensors to detect and report information security threats. HaaSS can detect, classify, and respond to various social engineering attacks and evaluate them against existing technical security mechanisms in a real-world context. Fan et al. [100] propose a model of human weakness based on human internal characteristics and the external circumstance influencing human nature for social engineering investigation. The authors categorize the defense measures into subjective (i.e., using standard security policies, updating facilities, and detecting malicious data) and objective (i.e., training human awareness, and detecting human emotion) to cope with human weaknesses.

5) DEFENSES TO ENSURE AVAILABILITY

The availability of communication buses, CPU, and memory for safety-critical applications in CAVs is essential [17]. Attacks, such as DoS, jamming, black hole, Sybil attacks, and vehicular malware or botnets can interfere with the transmission and routing of packets in CAVs, thus, affecting its overall availability [18]. The reliability of physical connections between nodes can also influence availability [101]. Physical availability attacks by blocking data between the sensors and the CAN network can be initiated using signal jamming [7]. With reference to Table 8, attack on system's availability can impact 03 04 06 07 21 shown in Figure 5 and Table 26 presents a synopsis of defenses to ensure availability.

TABLE 26. Defenses against to ensure availability.

Mechanism	Reference
Used real-time kinematic positioning-based receivers.	Cui et al. [102]
Software-based security framework.	Thangarajan et al. [103]
Hybrid communication architecture	Yastrebova et al. [104]
Signature-based authentication mechanisms	Lokman et al. [105]
Caching contents at different layers.	Liu et al. [9]
Encryption-based techniques.	Dibaei et al. [15]

The location solutions availability for CAVs is critical to achieving a 0.1 m real-time positioning accuracy. Cui et al. [102] investigate the positioning performance of Real-time kinematic positioning (RTK) based receivers. The authors report low-cost RTK receivers can address the communication link availability issues. Thangarajan et al. [103] present a lightweight software-based security framework for ECUs. The framework can provide diagnostics security solutions ensuring the availability of ECUs to deliver uninterrupted services like infotainment, telematics, diagnostics, and advanced driving assistance. Yastrebova et al. [104] propose a hybrid communication architecture to improve digital services availability to CAVs. The architecture exploits selected use cases and scenarios considering CAV viewpoints to improve road safety and create more efficient transport solutions.

Availability is an important requirement of the VANET to ensure that all the systems operate uninterruptedly [106]. The authors suggest defenses like signature-based authentication mechanisms, randomization of inter-arrival time, switching between channels, message-linkable group signatures, packet time-stamping, or leashes can be some potential solutions to prevent attacks affecting availability. Liu et al. [9] show that caching contents at different layers can be beneficial to improve the availability and achieve optimal performance in CAVs.

Also, encryption-based techniques can be applied for securing CAVs. Dibaei et al. [15] outlined symmetric key encryption, asymmetric key encryption, and attribute-based encryption to enhance vehicular network security for improving its availability. The authors describe how encryption can minimize Sybil attacks, replay attacks, DoS, eavesdropping, black hole attack, jamming, collusion, impersonation, and unauthorized access.

6) DEFENSES AGAINST DENIAL OF SLEEP

Energy depletion attacks involve disabling low-power (sleep) mode, increasing the amount of incoming or outgoing traffic, creating electromagnetic interference on wireless data transmission channels, and launching multiple applications or services of devices [40]. With reference to Table 9, denial of sleep attack can impact 03 06 10 22 23 shown in Figure 5 and Table 27 presents a synopsis of defenses against Denial of Sleep.

TABLE 27. Defenses against denial of sleep.

Mechanism	Reference
Time-division and channel hopping techniques.	Gallais et al. [41]
Zero-knowledge protocol-based authentication.	Naik et al. [107]
AI-based intelligent agents.	Udoh et al. [108]
Stochastic model monitoring sensor node behavior.	Bhattasali and Chaki [109]
Battery usage behavior analysis.	Desnitsky et al. [110]
Simplified authentication process. Hsueh et al. [111]	
Network organization and selective authentication.	Manju et al. [112]

Gallais et al. [41] investigate physical jamming scenarios preventing communications from taking place to cause retransmissions and additional duty of the target devices. Time division and channel-hopping techniques can mitigate jamming attacks. Naik et al. [107] address the denial of sleep attacks in WSN using a Zero-Knowledge Protocol (ZKP) to authenticate the sensor nodes that pass the sleep synchronization messages. Thus, every node sending the synchronization messages will be validated to accept the messages.

Udoh et al. [108] propose an architecture that propagates relevant knowledge via intelligent agents for mitigating denial-of-sleep attacks. The authors explain that each sensor can become an agent to sense data and take responsive action with the workload dynamically distributed among them. Bhattasali and Chaki [109] apply Absorbing Markov Chain (AMC) to model a sensor node behavior and detected denial of sleep attack by monitoring the entire network flow. Desnitsky et al. [110] analyze various types of battery depletion attacks on UAVs and their key characteristics. The authors design a prototype using Parrot AR-Drone to produce experimental results by simulating battery depletion attacks with and without physical contact.

Hsueh et al. [111] investigate shortcomings in media access control (MAC) protocols and suggested simplification of the authenticating process to reduce the energy consumption of sensor nodes. The authors propose a cross-layer design of a secure scheme integrating the MAC protocol, and their analysis shows that the scheme can counter the replay attack and forge attack without affecting the overall performance. Manju et al. [112] use network organization and selective-level authentication. The defending mechanism gets triggered only in the area of attack or when the attack is suspected, thus, communication overhead can be restricted.

7) DEFENSES AGAINST MALICIOUS CODE ATTACK

Malicious codes are a kind of traditional and common attack method for controlling the target [32]. Distributed DoS (DDOS) can send malicious messages using frequent transmissions and black hole or wormhole attacks can compromise routing protocols to threaten VANET availability [7], [113]. Malicious code attacks can target nodes to flood the network with a huge volume of dummy messages [70]. With reference to Table 10, malicious code attack can impact 02 05 06 07 11-13 16 18 21 shown in Figure 5 and Table 28 presents a synopsis of defenses against malicious code attacks.

TABLE 28. **Defenses against malicious code attack.**

Mechanism	Reference
Encryption, localization, and clustering mechanisms.	Dibaei et al. [15]
Firewall and self-isolation.	Thing et al. [114]
ML-based anomaly detection.	Wang et al. [115]
Convolutional neural networks-based approach.	Cui et al. [116], Van Wyk et al. [117]
Real-time data analysis method.	Park et al. [118]
Probabilistic modeling using feedback packet-based authentication techniques.	Abhishek et al. [119]
Collaborative detection strategy.	Wei et al. [120]

A malicious node can impersonate an RSU for tricking users to divulge the authentication details. Encryption, localization, and clustering mechanisms can be used to mitigate the impersonation attacks [15]. IP-based routing methods can prevent compromised ECU from performing malicious attacks on V2X. Similarly, firewalls can be employed to filter malicious messages from legitimate ones, or self-isolation of the systems can be introduced to prevent malicious code attacks [114].

Wang et al. [115] study sensor anomalies to recover the corrupt signals by utilizing the surrounding vehicles information. The authors explain that an anomaly in the data collected from a CAV sensor system can occur due to the presence of faulty sensors or malicious attacks. The approach employs an adaptive extended Kalman filter (AEKF) to smooth sensor readings of a CAV based on a nonlinear car-following model and detects sensor anomalies using One-Class Support Vector Machine (OCSVM) models based on the leading vehicle information. Cui et al. [116] propose a convolutional neural network (CNN)-based approach for the detection of malicious codes for a given network. The approach uses grayscale images obtained by converting executable files of malicious code as input to the CNN model. These grayscale images are created from executable binary files of malicious code by dividing them into an 8-bit length that can be converted to an unsigned integer number ranging from 0 to 255. Similarly, Van Wyk et al. [117] propose an anomaly detection approach using CNN and Kalman filtering with a χ^2-detector to detect and identify anomalous behavior in CAVs caused by faulty vehicle sensors or malicious cyber attacks. The authors evaluate the performance of the models to measure the overall proportion of correct predictions for normal and anomalous sensor values. Park et al. [118] propose a real-time data analysis method that can detect abnormal behaviors in large-scale network traffic due to malware.

Abhishek et al. [119] propose a lightweight mechanism called DRiVe that can detect malicious RSUs and can establish the data integrity for the CAVs. DRiVe incorporates a probabilistic model using feedback packet-based authentication techniques between a CAV and RSU to identify malicious RSUs. Wei et al. [120] design a collaborative detection strategy to detect malicious code by testing the runtime of identified tasks. Every task in CAVs can be time-bound and malicious operations will disrupt the execution time. The approach makes a distinction between normal and abnormal running scenarios to establish the presence of malicious code. Subsequently, the authors categorize the malicious code and analyzed its characteristics by simulating 554 IoT devices.

8) DEFENSES AGAINST INJECTION ATTACK

CAN bus, security protocol stack, on-board units, and sensors in CAVs can be a possible targets for injection attacks like OS command injection, HTML injection, or client-side template injection [15], [70], [72]. With reference to Table 11, injection attack can impact 03 04 08 18 20 shown in Figure 5 and Table 29 presents a synopsis of defenses against Injection attack.

TABLE 29. **Defenses against Injection attack.**

Mechanism	Reference
Cryptography-based defense mechanisms.	Aliwa et al. [7]
Sandboxing framework using a fault signature table.	Zhao et al. [121]
Diffusion function for non-invertibility and randomness.	Barbu et al. [122]
Run-time monitoring approach.	Cotroneo et al. [123]
Applied etiological and symptomatic approaches.	Mitropoulos et al. [124]
State-aware abnormal message injection attacks.	Xue et al. [125]
Intrusion detection model using feature generation and convolutional neural network-based approach.	Jeong et al. [126], Lokman et al. [105]

Aliwa et al. [7] describe potential defense mechanisms against injection attacks that include cryptography (e.g., cipher block chaining message authentication code), Hash message authentication code (e.g., symmetric key counters), and intrusion detection (e.g., anomaly-based methods). Zhao et al. [121] propose a sandboxing framework to detect the false data injection attack on CAVs. The framework leverages a fault signature table in diagnostics and develops a unique attack detection scheme. The authors evaluate the effectiveness of the approach using microscopic traffic simulation to detect the false data injection attack existing in the V2X communication.

Barbu et al. [122] propose a countermeasure framework against fault injection attacks to protect symmetric cryptosystem implementations. The framework relies on a good diffusion function having properties like non-invertibility and randomness that is achieved by using a Hash function with a counter, a chained block cipher, and a random linear function. Cotroneo et al. [123] propose a run-time monitoring approach for device drivers to detect I/O protocol violations due to incorrect commands injection or device state misinterpretation. Mitropoulos et al. [124] analyze defense mechanisms against web code injection attacks like cross-site scripting and SQL injection. The authors categorize the defense mechanisms as etiological, i.e., Parse-Tree Validation, Policy Enforcement, and Instruction Set Randomization; symptomatic, i.e., Taint tracking or training; and hybrid, i.e., combines characteristics from both etiological and symptomatic approaches.

Xue et al. [125] propose an approach that they call SAID to defend state-aware abnormal message injection attacks. SAID can detect the abnormal data to be injected into the in-vehicle network by considering the data semantics and the vehicle dynamics. Jeong et al. [126] propose an intrusion detection model based on feature generation and a convolutional neural network to prevent audio-video transport protocol (AVTP) stream injection attacks in automotive Ethernet-based networks. The authors evaluate their model using a physical BroadR-Reach-based testbed and captured real AVTP packets. Lokman et al. [105] investigate intrusion detection systems that use frequency-, machine learning-, statistical- and hybrid-based mechanisms for CAN bus network systems. The authors describe that factors like limited resources, the timing requirement, traffic patterns behavior, and unstable connection must be considered for designing the proposed solution in the CAN bus network system.

B. NETWORK SECURITY

Commonly used protocols in CAVs are CAN, LIN, MOST, DSRC, FlexRay, Automotive Ethernet, Bluetooth, Wi-Fi, and mobile 5G [7]. However, without adequate security mechanisms, connectivity expansion, such as V2V, V2R, or V2X can expose more attack surfaces and vulnerabilities. Thus, in overall CAVs, cyber-security risks, and vulnerabilities can arise from in-vehicle network attacks, vehicle-to-everything network attacks, and infrastructural- or slight attacks [14].

Multi-layer protection for network gateways can be achieved by applying domain isolation principles, i.e., separating interfaces for safety-critical systems from non-critical systems [127], and least privilege principle, i.e., given minimal access or permissions to execute tasks [128]. Considering that external data is potentially hostile, it must be properly validated and scanned [129]. Moreover, any abnormal communication or messages that deviate from predefined behaviors can be restricted proactively, i.e., allow communications and messages only between pre-approved systems and sensors, block unapproved and malicious messages, and alert security systems about any invalid attempts. Azees et al. [130] survey the security vulnerabilities and proposed the countermeasures for VANETs.

1) DEFENSES AGAINST MiTM

Jasek et al. [129] suggest that data transmission between CAVs and RTUs must authenticate each other and be properly encrypted to minimize MiTM attacks. Conti et al. [45] categorize MiTM attacks based on attack characteristics, i.e., impersonation techniques, the communication channel in which the attack is executed, and the location of the attacker and target in the network. The authors present potential MiTM defense mechanisms according to used approaches and context (abstract layer) of applicability for the four categories of MiTM attacks described in Section IV-B. With reference to Table 12, MiTM attack can impact 03 05 06 09 21 shown in Figure 5 and Table 30 presents a synopsis of defenses against MiTM.

TABLE 30. **Defenses against MiTM.**

Mechanism	Reference
Intrusion Detection System.	Aliyu et al. [131]
Transport Layer Security security-enhanced protection mechanism.	Yang et al. [132]
Cryptographic, voting-based solutions, IP filtering, certificate pinning approach, robust authentication protocols, and cipher algorithms.	Conti et al. [45]

Aliyu et al. [131] propose an Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) for MiTM attacks at the Edge layer. The authors apply Advanced Encryption System (AES) symmetric encryption technique using Diffie-Hellman key exchange. However, Shen et al. [133] describe that the lack of mutual authentication can make the Diffie-Hellman key exchange vulnerable to the MiTM attack. Consequently, the authors propose an in-band solution during the key establishment process for wireless devices to prevent MiTM attacks. The protocol forces a successful MiTM attacker to cause consecutive packet collisions at the link layer that can be detected by the proposed attacker detection algorithm by distinguishing the consecutive packet collision introduced by the MiTM attacker from normal packet collisions. Yang et al. [132] propose a Transport Layer Security security-enhanced protection mechanism (TLSsem) to enhance server authentication by combining client authentication with TLS session establishment. TLSsem involves pre-binding, certificate validation, and port hopping to ensure the reliability of wireless communications.

2) DEFENSES AGAINST EAVESDROPPING OR SNIFFING

Anonymization, resource management, trust-based recommendation, or scheduling mechanism can be potential defense strategies for eavesdropping or sniffing attacks [14]. With reference to Table 13, eavesdropping or sniffing attack can impact 05 06 21 shown in Figure 5 and Table 31 presents a synopsis of defenses against Eavesdropping or Sniffing.

TABLE 31. **Defenses against eavesdropping or sniffing.**

Mechanism	Reference
Elliptic Curve Cryptography (ECC).	Chhabra and Arora [134]
Use of a using a jamming signal to avoid an eavesdropper.	Choi et al. [135]
Lattice-based network coding signature scheme.	Wu et al. [136]
Immunizing coding.	Li et al. [137]

Chhabra and Arora [134] propose Elliptic Curve Cryptography (ECC) based scheme by fragmenting the data and then pseudo-randomly allotting the different data packets for securing the servers against eavesdroppers. Choi et al. [135] propose a security-enhancing transmission scheme using a jamming signal when an eavesdropper is near the receiver and has a correlated channel. The authors demonstrate that mixing the desired signal with the jamming signal can be an effective solution, as it degrades the signal quality received by the eavesdropper.

Wu et al. [136] propose a lattice-based network coding signature scheme to blind the global coefficient matrix by encrypting the original encoding vector. The authors provide proof that the scheme is secure against both eavesdropping and pollution attacks assuming the pseudo-random function and the small integer solution (SIS) are applied to secure the standard lattices. Li et al. [137] implement the immunizing coding (iCoding) method to prevent interference and eavesdropping in Wireless Communications. An iCoded signal is generated and sent by the legitimate transmitter (Tx) by exploiting both channel state information (CSI) and data carried in the interference. The iCoded signal interacts with the interference at the desired/legitimate Receiver (Rx) ensuring intended data can be recovered achieving interference-free desired transmission. Liao et al. [138] utilize the (n, k) erasure coding with network coding to tackle the eavesdropping problem in heterogeneous IoT systems.

3) DEFENSES AGAINST DENIAL OF SERVICE

DoS attacks can occur at various layers of a computer system, including the hardware, network, and software layers. However, the most common and well-known type of DoS attack is a network-layer DoS attack. In CAVs, a DoS attack can target several components like sensors, Electronic Throttle Control Systems (ETC), CAN bus, or infotainment systems [47]. Cao et al. [139] survey countermeasures against DoS attacks including event-triggered control systems, probability measurement of stochastic processes using the Markov model, Queuing model, or Bernoulli model to characterize system properties under DoS attacks. An anomaly-based detection mechanism can be employed to protect nodes against DoS attacks on wireless sensors [15]. With reference to Table 14, denial of service attack can impact 05 06 21 22 shown in Figure 5 and Table 32 presents a synopsis of defenses against Denial of Service.

TABLE 32. Defenses against denial of service.

Mechanism	Reference
Penetration test methods.	Andreica et al. [140]
Key management scheme.	Nanda and Krishna [141]
Prediction-Based Authentication.	Lyu et al. [142]
Reputation framework.	Tian et al. [143]
Node behavior using cooperative tracking.	Wang et al. [144]
Anomaly detection by deriving quantitative relationships.	Zhang et al. [145]
Anomaly detection by tracking the delays in information processing.	Biron et al. [146]
Incentive scheme for reliable cooperative downloading and forwarding.	Lai et al. [147]

Andreica et al. [140] propose penetration test methods for DoS and MiTM attacks using the User Datagram Protocol (UDP) and the Teltonika protocol, respectively for GPS-based monitored intelligent transportation systems. Nanda and Krishna [141] propose a key management scheme using the concept of timestamp and delay to protect the server in a hierarchical sensor network from a DoS attack. Lyu et al. [142] propose Prediction-Based Authentication (PBA), i.e., a broadcast authentication scheme for securing vehicle-to-vehicle communications. PBA can defend against computation-based DoS attacks as well as resist packet losses caused by the high mobility of vehicles.

Tian et al. [143] design a reputation framework (VCash) that combines entity-centric- and data-centric methods to identify the denial of traffic service. VCash can also restrict the spread of false messages, and encourage the contribution of traffic condition monitoring and verification. Wang et al. [144] suggest a cooperative secure control approach in the presence of intermittent DoS attacks. The cooperative tracking objective exploits the topology-dependent Lyapunov function method and topology-allocation-dependent average dwell-time (TADADT) scheme, to achieve the convergence of estimation errors and the coordination tracking based on the output information obtained from each node. Zhang et al. [145] describe system design conditions by deriving quantitative relationships between attack parameters and system performance using time-varying sampling and the simulation of a DoS attack.

Biron et al. [146] propose a real-time scheme for the diagnosis of DoS attacks by modeling the effect of the attack by a time delay in the information processing via a communication network. The main objective of the scheme is to track the delay in information processing using a set of observers based on sliding mode theory and adaptive observer theory. Lai et al. [147] design a secure incentive scheme for reliable cooperative downloading and cooperative forwarding in Vehicular Ad hoc Networks (VANETs) by applying virtual checks, i.e., associated with the designated verifier signature to ensure fair and secure cooperation. The scheme together with the single pruning search (SPS) or paired single pruning search (PSPS) method can detect and weaken denial of service (DoS) attacks.

4) DEFENSES AGAINST ROUTING-, BLACK HOLE-, GRAY HOLE-, AND WORMHOLE ATTACK

Routing involves the identification, selection, and establishment of the best shortest path for message communication [148]. Routing information can be altered by redirecting or dropping data packets at the communication level [149]. Thus, the malicious nodes can behave like *black holes*, i.e., drain all network packets, *gray holes*, i.e., drain selective packets, or *wormholes*, i.e., migrate packets from one network location to another after recording the packets. A wormhole is a type of DoS attack that can compromise the routing protocols by creating a tunnel between two or more malicious entities to transmit data packets [106]. Butan et al. [31] further show that routing attacks can be originated due to misdirection, network partitioning, routing loop creation, and spoofed, altered, or replayed routing information. With reference to Table 15, 16, and 17, routing-, black hole-, gray hole-, and wormhole attack can impact 03-05 06 08 20 21 shown in Figure 5 and Table 33 presents a synopsis of defenses against Routing-, Black hole-, Gray hole-, and Wormhole attack.

Some of the potential solutions for countering routing attacks can be ant colony optimization, swarm algorithms of artificial intelligence, variable control chart, and trust calculation [14]. Nayak et al. [150] propose a deep learning-based

TABLE 33. Defenses against Routing-, Black hole-, Gray hole-, and Wormhole attack.

Mechanism	Reference
Ant colony optimization, swarm algorithms, variable control chart, and trust calculation.	Sun et al. [14]
Deep learning-based routing attack detection mechanism.	Nayak et al. [150]
Intrusion detection scheme.	Wazid et al [151]
Attack location tracing using a probabilistic packet marking model.	Zhao and Dong [152]
Selective packet drops detection algorithm.	Shu and Krunz [153]
Multiplicative sensor watermarking scheme	Ferrari and Teixeira [154]
Broadcast alarm, alternative routing path, blacklisting nodes.	Wang et al. [155]
ActiveTrust.	Liu et al. [156]
Statistics approach for observing discrepancies and route backtracking mechanism.	Tobin et al. [157]
Denial contradictions.	Schweitzer et al. [158]
Covert in-band channel creation.	Hua [159]
Intrusion detection system	Bhosale and Sonavane [52]
Cross-layer verification framework.	Jagadeesan and Parthasarathy [160]

routing attack detection mechanism for IIoT that can discriminate between real and misleading data, detect an attack event, and classify the attack types into corresponding classes involved in Low-Power and Lossy Networks (RPL) routing. The authors apply adversarial training of the model for detecting intended attacks in routing protocol in RPL. Wazid et al. [151] propose an intrusion detection scheme, RAD-EI, to detect routing attacks in an edge computing-based IoT environment. The authors perform RAD-EI security analysis and simulation using NS2 simulation against malicious routing attacks. Zhao and Dong [152] propose a method that can trace the location of the attack source actively after the DOS attacks are found in the destination node using a probabilistic packet marking model. Ren et al. [72] briefly describe some of the protocol verification tools, e.g., *Tamarin*, *ProVerif*, *Verifpal*, *CryptoVerif*, *Scyther*, *AVISPA*, that can be utilized for formal analysis of security of protocols.

Shu and Krunz [153] propose an algorithm to detect selective packet drops made by insider attackers that are based on detecting the correlations between the lost packets over each hop of the path. The authors also design a homomorphic linear authenticator (HLA) based public auditing architecture that allows the detector to verify the truthfulness of the packet loss information reported by nodes ensuring truthful calculation of their correlations algorithms. Ferrari and Teixeira [154] propose a multiplicative sensor watermarking scheme to separately watermark each sensor output by a Single Input Single Output (SISO) filter. The authors analyze the physical sensor re-routing attack and the cyber measurement re-routing one that can be leveraged to detect cyber sensor routing attacks.

Wang et al. [155] discuss security issues caused by black- and gray-hole attacks in the V2X network. The authors analyze reputation-based, acknowledgment (ACK)-based, and detection-based methods and summarized the prevention methods. Liu et al. [156] propose a security and trust routing scheme, ActiveTrust, based on an active detection that can establish nodal trust by detecting suspicious nodes. The ActiveTrust scheme exploits residue energy to construct multiple detection routes that improve the data route success probability and ability to counter black hole attacks as well as optimize network lifetime. Tobin et al. [157] apply a route backtracking mechanism and observed discrepancies in statistics reported by intermediate nodes for black hole attack detection. The authors run a simulation using Network Simulator-3 (NS-3) to evaluate their scheme on a VANET.

Schweitzer et al. [158] implement denial contradictions with a fictitious node (DCFN) mechanism for minimizing the gray-hole DoS attack. The technique is evaluated for five different attack scenarios, i.e., passive silent attack, randomly located attack, initially one-hop neighbor attack, shadow attack, and MiTM attack. Doshi et al. [161] propose a game-theoretic approach to prevent gray-hole attacks in wireless ad hoc networks. Each node can implement the suggested strategies without additional network-level overhead to counter the attack.

Wormhole attacks can be countered using bound distance or time and graph theories or geometry-based mechanisms. Hua et al. [159] propose an approach using a relay host to build a covert in-band channel between the two compromised switches. The authors exploit Mininet 3.3.0d4, OpenFlow 1.5, Open vSwitch 2.11.0, and FloodLight controllers to emulate the wormhole attacks and compare the performance of their proposed method with different compromised switch pairs. Bhosale and Sonavane [52] design and implement an intrusion detection system to detect wormhole attacks using Contiki OS and Cooja Simulator. Jagadeesan and Parthasarathy [160] propose a cross-layer verification framework that can detect and counter black hole and wormhole attacks in wireless ad-hoc networks.

C. HARDWARE SECURITY

CAVs can be specified as specialized physical nodes or devices that are capable of *sensing*, *communicating*, *computing*, and *storage* [9] and are far more vulnerable than a typical IoT node.

1) DEFENSES AGAINST NODE-CAPTURE

A node-capture attack can be a major concern for the CAV ecosystem. Controlling a CAV by reprogramming and redeploying it can be hazardous for the entire CAV ecosystem [162]. With reference to Table 18, node-capture attack can impact 03 07 08 11-15 18 20 shown in Figure 5 and Table 34 presents a synopsis of defenses against node-capture.

TABLE 34. Defense against node-capture.

Mechanism	Reference
Honeywords technique and the fuzzy-verifier.	Li et al. [163]
Secure random key distribution (SRKD) scheme.	Li et al. [164]
Q-composite scheme against node capture.	Zhao [165]
RNN (recurrent neural network)-based detection.	Gu et al. [166]
Program integrity verification (PIV) protocol.	Agrawal et al. [53]
key agreement scheme using Exclusion Basis Systems structure.	Zhang and Li [167]

Wang et al. [168] investigate the causes and the consequences of node-capture attacks. The authors categorize node-capture attacks based on attack surface, vulnerabilities, and adversary capabilities. They further elaborate on each type of attack by examining vulnerable protocols to investigate the potential countermeasures. Li et al. [163] adopt the honeywords technique and the fuzzy-verifier to counter node capture attacks. The authors revisit and evaluate forty-two representative schemes to determine the desirable attributes and security requirements for IIoT-based authentication schemes.

Li et al. [164] propose a secure random key distribution (SRKD) scheme that has shown a higher resilience against the other random key distribution schemes used for preventing node capture and information eavesdropping. The authors describe that SRKD can be applied against the node replication attack and can prevent replication nodes from injecting false information. Zhao [165] evaluates the resilience of the q-composite scheme against node capture. Gu et al. [166] propose an RNN-based detection method that can be used to detect node capture in wireless sensor networks (WSNs).

Agrawal et al. [53] propose a program integrity verification (PIV) protocol that can detect if a captured node is redeployed in a cluster. Each cluster head managing a group of nodes in the network can be equipped with trusted platform module (TPM) capabilities. A cluster head can serve as a TPM-enabled verification server (TVS) to check if a redeployed node is a victim of node capture by verifying the integrity of the node program. Zhang and Li [167] propose a key agreement scheme using the Exclusion Basis Systems (EBS) structure to update the group keys between clusters. The simulation experiments performed by the authors have shown the scheme is capable of good connectivity and resistance against node capture.

2) DEFENSES AGAINST SIDE-CHANNEL ATTACK

Side-channel attacks are typically timing attacks, power monitoring attacks, electromagnetic leaks, acoustic signals, transient characteristics, or data remanence at hardware or software levels [169], [170]. Le et al. [17] explain reliance on obscurity for security, bad programming practices, and lack of cryptographic- or insufficient hardware protection can cause side-channel attacks affecting critical devices like ECUs or V2X communications. With reference to Table 19, side-channel attack can impact 03 05 06 11-13 18 19 21 shown in Figure 5 and Table 35 presents a synopsis of defenses against side-channel attack.

Kiaei et al. [171] introduce a processor, SKIVA, that can protect ciphers against timing-based side-channel analysis, power-based side-channel analysis, and/or fault injection at various levels of security. Alwarafy et al. [149] suggest de-patterning data transmissions by intentionally inserting fake packets that change the traffic pattern to prevent side-channel attacks. Lavaud et al. [177] survey the class of side-channel attacks that include both non-electromagnetic or electromagnetic and their countermeasures.

TABLE 35. Defenses against side-channel attack.

Mechanism	Reference
Secure processor.	Kiaei et al. [171]
Simulation technique by inserting fake packets.	Alwarafy et al. [149]
Cipher block chaining encryption.	Wang et al. [172]
CAD tool framework.	Park et al. [173]
Dynamically adjusts the granularity of platform time sources.	Liu et al. [174]
CacheFix.	Chattopadhyay and Roychoudhury [175]
End-to-end static analysis tool.	Chen et al. [176]

Wang et al. [172] study the cryptographic operation of SSL/TLS Record Protocol in cipher block chaining (CBC)-mode encryption. Park et al. [173] propose a CAD tool framework for automatic timing attack vulnerability evaluation and associated algorithms and metrics at the early design stage. The framework can automatically identify timing variance-induced side-channel attacks in a Register Transfer Level (RTL) design for the FPGA-based design flow. Liu et al. [174] propose a technique that dynamically adjusts the granularity of platform time sources to periodically mitigate side-channel attacks. The method allows virtual machines (VMs) to dynamically request the time stamp counter (TSC) on the platform, i.e., the VM application sends on-demand requests to the hypervisor to mask low-order bits of the TSC to disable precise time measurements by another co-resident VM. The authors evaluate their methods against covert-channel timing attacks like the Last Level Cache (LLC) attack and memory bus contention attack.

If a given software, e.g., an encryption routine, satisfies cache side-channel freedom, it can be asserted that the software can address cache timing attacks. Chattopadhyay and Roychoudhury [175] propose CacheFix which verifies the cache side-channel freedom of an arbitrary program. The framework automatically builds and refines the abstraction of cache semantics that includes direct-mapped caches, set-associative caches with the least recently used (LRU), and first-in-first-out (FIFO) policy. The core symbolic engine of CacheFix can systematically combine its reasoning power with runtime monitoring to ensure cache side-channel freedom during program execution that was evaluated on 25 routines from actual cryptographic libraries. Chen et al. [176] propose an end-to-end static analysis tool for finding resource-usage side-channel vulnerabilities in Java applications that they call Themis. The tool uses Quantitative Cartesian Hoare Logic (QCHL) to verify ϵ-bounded noninterference to detect non-trivial vulnerabilities in real-world Java programs.

3) DEFENSES AGAINST TIMING ATTACK

Timing attacks in Intelligent Transportation Systems (ITS) can delay the transmission of security messages, which can lead to accidents [70]. With reference to Table 20, timing attack can impact 03-05 06-08 11-13 18 shown in Figure 5 and Table 36 presents a synopsis of defenses against timing attack.

TABLE 36. Defenses against timing attack.

Mechanism	Reference
Packet leash mechanism.	Benzarti et al. [178]
Real-time monitoring mechanism.	Lavaud et al. [177]
Formal verification of timed security protocols.	Li et al. [179]
Workflow verification.	Li et al. [180]
Behavioral tests.	Selis and Marshal [181]
Benchmarking applications against known standards.	Peter and Givargis [182]
Task-offloading schemes.	Meng et al. [183]
Dedicated tool to observe timing attacks.	Javeed et al. [184]

Timing attacks targeting the execution time of cryptographic instructions can be prevented using packet leashes mechanism [178] or constant-time security where every security-critical operation is monitored [177]. Li et al. [179] propose timed applied p-calculus as a formal language for specifying timed security protocols. The authors define its formal semantics based on timed logic rules that can facilitate efficient verification against various authentication and secrecy properties. Security Protocol Analyzer (SPA) is implemented using the method that can analyze a wide range of protocols, e.g., Wide Mouthed Frog (WMF), Kerberos-, distance bounding-, Needham-Schroeder, International Telegraph and Telephone Consultative Committee (CCITT), Secure Key Exchange Mechanism (SKEME) protocols.

Li et al. [180] define a workflow to verify the timing attack process and optimize the attack steps from the perspective of the security of the time side channel. Selis and Marshal [181] analyze a fake timing attack against behavioral tests, i.e., extracting timing information from a system using a characterization algorithm based on pinging localhost. The authors propose an algorithm for detecting the attack including forged embedded machines based on virtual and emulated systems to create trusted M2M communications. Peter and Givargis [182] investigate the integration of timing attack resilience into the high-level synthesis (HLS) that can be expressed in higher-level programming languages. The authors use a low-level virtual machine compiler back-end and a scalable data frame compiler scheduler approach in the open-source HLS tool LegUp to benchmark applications, such as cryptographic standards ECC and RSA.

Meng et al. [183] show that task-offloading can be particularly vulnerable to timing attacks due to frequent sending/receiving. The authors propose an offloading scheme that combines regular rekeying and random padding. Their experiments show the attacker needs more samples to conduct a timing attack when random padding is deployed in the system. Javeed et al. [184] design a tool that can monitor the time readings at runtime on a per-process basis. The authors evaluate their approach using five timing attacks, i.e., *Meltdown*, *Evict+Reload*, *Flush+Flush*, *Flush+Reload*, and *Prime+Probe*.

4) DEFENSES AGAINST CRYPTANALYSIS ATTACK

In cryptanalysis attacks, the attacker focuses on bypassing or breaking cryptographic security mechanisms [16]. Cryptanalysis attacks can be categorized as a *known-plaintext attack*, *chosen-plaintext attack*, *ciphertext-only attack*, *chosen-ciphertext attack*, and *chosen-key attack* considering the attacker extracted the encryption key by exploiting either plaintext or ciphertext [185]. Chattopadhyay et al. [186] explain that decoding application protocol messages, reverse-engineering security-critical parameters, or cryptanalysis of cryptograms are the typical ways for orchestrating cryptanalysis attacks. Sravani and Durai [187] study cryptanalysis-based strategies to attack crypto-devices using side-channel and hardware trojan techniques. With reference to Table 21, Cryptanalysis attack can impact 02 04 05 06 07 18 20-22 shown in Figure 5 and Table 37 presents a synopsis of defenses against Cryptanalysis attacks.

TABLE 37. Defenses against cryptanalysis attack.

Mechanism	Reference
Block cipher-based encryption.	Babu & Kumar [188]
Use of IDS and firewalls.	Aliwa et al. [7]
Mathematical models and frameworks of physical layer encryption.	Li et al. [189]
EmuLab for evaluating Security Credential Management System.	Mushrall et al. [190]

Babu and Kumar [188] propose an SMS4-BSK cryptosystem that is implemented using a Kintex 7 FPGA. SMS4-BSK is a block cipher that provides faster encryption by dividing message signal, i.e., plaintext into blocks of 128 bits. It can resist cryptanalysis over the ciphertext-only attack, known-plaintext attack, chosen-plaintext attack, and chosen-ciphertext attack. Cryptography-based methods, IDS, and firewalls can secure the in-vehicle networks from malicious messages and their detection and blocking access to the internal buses [7].

Li et al. [189] focus on the mathematical models and frameworks of physical layer encryption (PLE) to establish cryptographic primitives for PLE summarizing the basic design rules. The authors evaluate their schemes, i.e., isometry-based block PLE- and stream PLE framework to resist known-plaintext attacks (KPAs) and chosen-plaintext attacks (CPAs) and compare the performance with existing schemes, such as phase rotation scheme, intrinsic interference scheme, and sub-carrier obfuscate and dummy. Mushrall et al. [190] design EmuLab that can facilitate research to strengthen Security Credential Management System (SCMS). SCMS implements Vehicular Public Key Infrastructure (V-PKI) to provide digital certificates. Two sets of experiments, i.e., secure vehicular communications with pseudonym certificates and overhead of SCMS self-healing from an incident of compromising a Root Certificate Authority (CA) were used to demonstrate the working of EmuLab. The authors describe that EmuLab can not only support academic and industrial research on high-risk projects to secure vehicular communications but also strengthen SCMS against post-quantum cryptanalysis.

TABLE 38. Attacks vs. security properties.

	Attack Type	Availability	Authentication	Confidentiality	Integrity	Non-repudiation	Privacy	Impacts and outcomes	Reference Architecture 5
Software	Fake node or Sybil	✓	✓					Communication and coordination between vehicles, false data transmission, traffic flow disruption, access to sensitive information.	01 04 05 06 09 11-16 20-22
	Replay		✓		✓	✓		Legitimate communication interception and recording	03-05 06 07 16 20-22
	AI/ML	✓		✓	✓			Vehicle's sensor data manipulation, traffic signs manipulations, data falsification, false driving maneuver, steal sensitive data.	01 04 08 09 17 20
	Social engineering		✓				✓	Compromise authentication mechanisms, login credentials, and sensitive data theft.	04 05 07 09 14 18 20
	System's availability	✓						Denial of service, jamming, degrading the data processing ability	03 04 06 07 21
	Denial of Sleep	✓						Energy depletion, erratic behavior or loss of control.	03 06 10 22 23
	Malicious Code	✓		✓	✓			Safety risks, privacy breaches, financial losses.	02-05 06 07 11-13 16 18 21
	Injection attack			✓	✓	✓		Message fabrication and alteration, data tempering and suppression.	03 04 08 18 20
Network	Man-in-The-Middle		✓	✓	✓	✓		Data theft and manipulation, Unauthorized access, service disruption.	03 05 06 09 21
	Eavesdropping or sniffing		✓	✓			✓	Data theft and loss of critical business information, unauthorized access, privacy violation.	05 06 21
	Denial of Service	✓	✓					System failure, loss of control, delayed response time, financial losses, reputation damage.	05 06 21 22
	Routing	✓	✓		✓			Disruption of communication, spoofing, end-to-end delay, affect uptime, and increase error messages	03-05 06
	Black hole, Gray hole, Wormhole	✓	✓		✓			Disruption of communication, unauthorized access, data manipulation, inaccurate routing.	03-05 06 08 20 21
Hardware	Node-capture	✓		✓	✓			Communication protocols, boot processes, firmware updates, vehicle to malfunction or crash, data modification or tampering, sensitive information such as location, personal identification, and operational details.	03 07 08 11-15 18 20
	Side-channel	✓		✓	✓			Power consumption, electromagnetic radiation, or timing information, infer sensitive information like encryption keys.	03 05 06 11-13 18 19 21
	Timing	✓		✓	✓			Cryptographic operation, location tracking, remote control, data theft such as login credentials or other authentication tokens.	03-05 06-08 11-13 18
	Cryptanalysis	✓	✓	✓	✓			Unauthorized access and controlling functions like steering, braking, and acceleration, data theft and manipulation, malware injection, and disruption the communication.	02 04 05 06 07 18 20-22

VI. DISCUSSIONS AND POTENTIAL RESEARCH DIRECTION

In this section, we present a mapping between attacks and security properties that can be useful for understanding the impacts and outcomes based on the findings in the previous sections. Subsequently, we discuss research trends and directions for a safe and secure CAVs ecosystem.

A. ATTACKS VS. SECURITY PROPERTIES

We discuss software, network, and hardware security measures that can be useful for securing a CAV ecosystem. Table 38 analyzes the effect of twenty-one attack on integral security properties, such as availability, authentication, integrity, confidentiality, non-repudiation, and privacy. Availability can be impacted if legitimate users are denied from accessing the system. Any disruption to authentication mechanisms can prevent legitimate users from accessing the system while allowing adversaries to gain unauthorized access. The integrity of a CAV system can be affected if an attack leads to the loss, alteration, or destruction of data. The confidentiality of a CAV system can be impacted if legitimate users are prevented from accessing the system and at the same time adversaries gain unauthorized access and steal sensitive information. An attack can affect the non-repudiation aspect of a CAV system by preventing the system from logging and recording the actions of legitimate users, making it difficult to prove who performed certain actions in the system. Finally, the privacy of CAV users can be compromised if their personal and sensitive information got exposed.

B. RESEARCH TRENDS AND DIRECTIONS

The rapid development of a reliable CAVs ecosystem is essential for smart transportation, it can offer more safety and comfort on roads, prevent and mitigate accidents, and reduce greenhouse gas emissions and energy utilization. Figure 7 illustrates prominent companies such as *Waymo*, *Tesla*, *Toyota*, *General Motors*, *Nissan*, *Ford*, *Uber*, *Baidu* and *Wayve* that are heavily investing on the research and development of CAVs across the world [191]. Many of CAVs manufacturers such as Waymo, Toyota, Nissan, and General Motors have joined the Auto-ISAC, an industry-operated initiative created to increase cybersecurity awareness and collaboration across the global automotive industry [192]. Auto-ISAC enables manufacturers to share and analyze cybersecurity incidents, threats, and violations to collectively enhance vehicle cybersecurity capabilities across the global automotive industry.

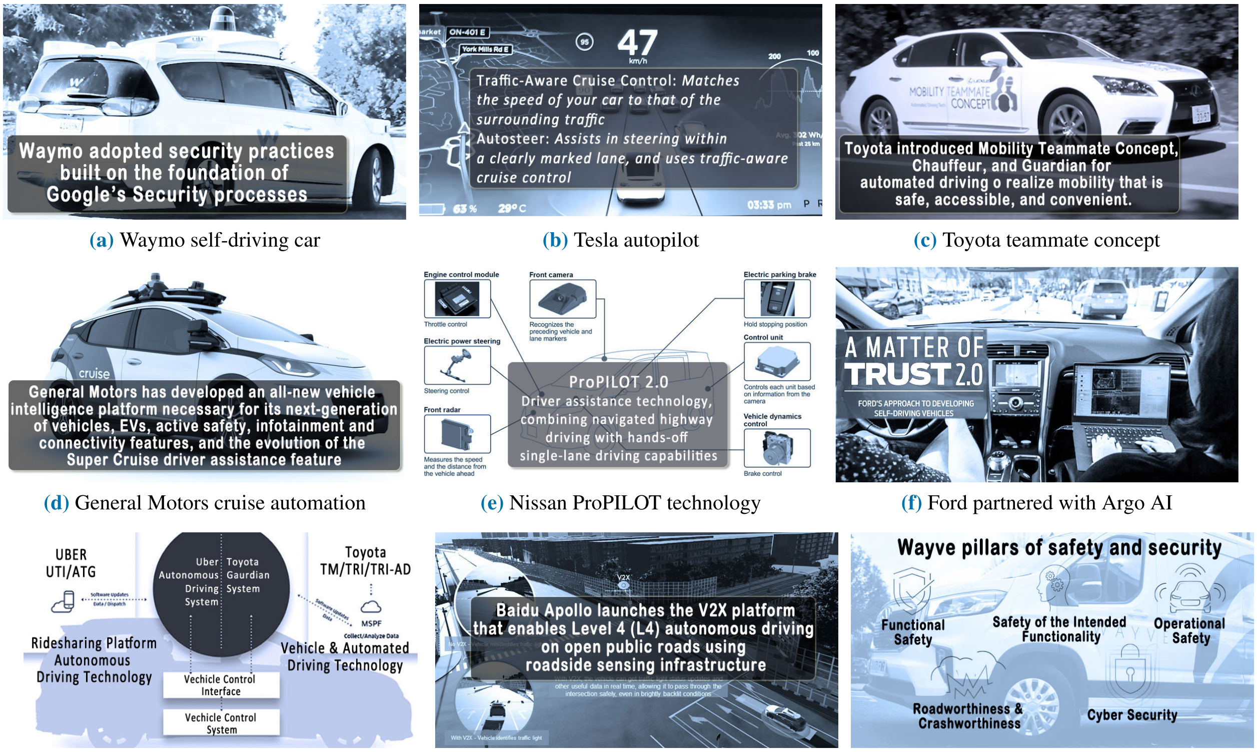
Waymo can be considered a leading company in the field of self-driving technology. Waymo adopted security practices built on the foundation of Google's Security processes [193]. Toyota introduced Mobility Teammate Concept, Chauffeur, and Guardian for automated driving to realize mobility that is safe, accessible, and convenient. Toyota engages with security researchers and other vehicle manufacturers regarding vehicle and enterprise cybersecurity [194]. Toyota has partnered with Tencent Keen Security Lab to strengthen the security functions of connected vehicles and enhance road safety. Uber has been working on developing self-driving cars alongside Toyota Guardian technology to supplement Uber's ride-sharing service and sharing network.

Tesla has been working on developing its self-driving technology called Autopilot. General Motors has been investing in developing autonomous vehicle technology through its subsidiary, Cruise Automation. Nissan has developed ProPILOT technology for a semi-autonomous driving system and developed vision-based advanced driver-assistance systems (ADAS) to accelerate the development of its autonomous vehicle technology. Wayve is developing its own AI-driven self-driving technology. Wayve's safety framework focused on five distinct safety areas, i.e., functional safety, the safety of the intended functionality, operational safety, crashworthiness, and cyber security to design, test and deploy autonomous driving technology [195].

Ford Motor Company announced the formation of a new subsidiary in 2021, Ford Autonomous Vehicles LLC, to focus on its autonomous vehicle efforts. Ford partnered with Argo AI to build high-quality self-driving vehicles [8]. NXP and the Ford Motor Company partnered to deliver next-generation connected car experiences and expanded services across the global fleet of vehicles [196]. NXP's vehicle network processors can provide secure, in-vehicle networking and enable the gateway to rapidly deploy Over-the-Air (OTA) software updates and new services. Baidu has been developing its autonomous driving technology, called Apollo, for several years. Baidu Apollo launches the V2X platform that enables Level 4 (L4) autonomous driving on open public roads using roadside sensing infrastructure [197]. The state of Michigan and the Cavnue consortium build a special corridor on I-95 between Detroit and Ann Arbor for CAVs [198]. CAVs operating in these lanes can use the information on their surroundings and communicate with each other to move faster, more safely, and at closer distances, allowing more capacity in the same space.

ISO/SAE 21434 is the first standard that will be jointly released by both SAE and ISO under the new agreement [199]. ISO/SAE 21434 defines a structured process to ensure cybersecurity is incorporated upfront to minimize the possibilities of attacks, thus reducing the likelihood of losses [200]. Further, the structured process provides a clear means to react to a continually changing threat landscape, maintains consistency across the global industry, and promotes a complete and conscious decision-making process.

As illustrated in Figure 8, cybersecurity activities/processes cover all phases of the vehicle lifecycle, i.e., design and engineering, production, operation by the customers, maintenance and service, and decommissioning [201]. ISO/SAE 21434 applies to all road vehicles including underlying systems, components, software, and connection from the vehicle to external devices/networks. It suggests following a risk-oriented approach, the risk is used for prioritization of action and analysis of risk factors for methodical elicitation of cybersecurity requirements.


Waymo adopted security practices built on the foundation of Google's Security processes
47 km/h
Traffic-Aware Cruise Control: Matches the speed of your car to that of the surrounding traffic
Autosteer: Assists in steering within a clearly marked lane, and uses traffic-aware cruise control
Toyota introduced Mobility Teammate Concept, Chauffeur, and Guardian for automated driving o realize mobility that is safe, accessible, and convenient.
MOBILITY TEAMMATE CONCEPT
cruise
General Motors has developed an all-new vehicle intelligence platform necessary for its next-generation of vehicles, EVs, active safety, infotainment and connectivity features, and the evolution of the Super Cruise driver assistance feature
Engine control module
Throttle control
Front camera
Recognizes the preceding vehicle and lane markers
Electric parking brake
Hold stopping position
Electric power steering
Steering control
Control unit
Controls each unit based on information from the camera
Front radar
Measures the speed and the distance from the vehicle ahead
Vehicle dynamics control
Brake control
ProPILOT 2.0
Driver assistance technology, combining navigated highway driving with hands-off single-lane driving capabilities
A MATTER OF TRUST 2.0
FORD'S APPROACH TO DEVELOPING SELF-DRIVING VEHICLES
UBER UTI/ATG
Toyota TM/TRI/TRI-AD
Uber Autonomous Driving System
Toyota Gaurdian System
Software updates
Data / Dispatch
MSPF
Collect/Analyze Data
Ridesharing Platform Autonomous Driving Technology
Vehicle & Automated Driving Technology
Vechicle Control Interface
Vehicle Control System
Baidu Apollo launches the V2X platform that enables Level 4 (L4) autonomous driving on open public roads using roadside sensing infrastructure
Wayve pillars of safety and security
Functional Safety
Safety of the Intended Functionality
Operational Safety
Roadworthiness & Crashworthiness
Cyber Security

(a) Waymo self-driving car (b) Tesla autopilot (c) Toyota teammate concept

(d) General Motors cruise automation (e) Nissan ProPILOT technology (f) Ford partnered with Argo AI

(g) Uber integrate its self-driving technology alongside Toyota Guardian technology (h) Apollo - Baidu autonomous driving technology (i) Wayve pillars of safety and security

FIGURE 7. Companies leading the research and development of connected and autonomous vehicles across the world.

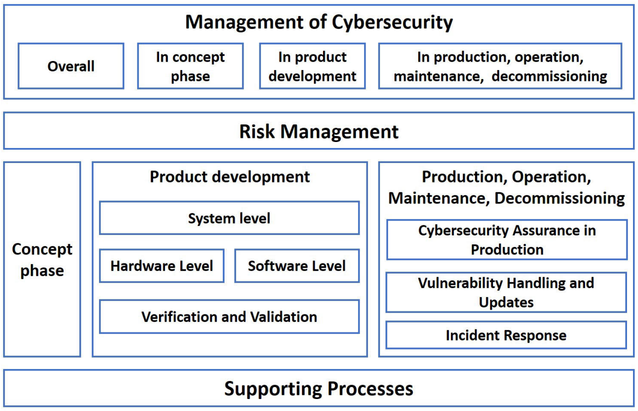
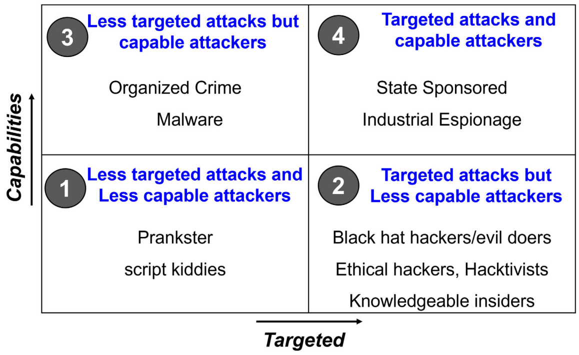

Management of Cybersecurity
Overall
In concept phase
In product development
In production, operation, maintenance, decommissioning
Risk Management
Concept phase
Product development
System level
Hardware Level
Software Level
Verification and Validation
Production, Operation, Maintenance, Decommissioning
Cybersecurity Assurance in Production
Vulnerability Handling and Updates
Incident Response
Supporting Processes

FIGURE 8. An overview to ISO21434 structure that describes the security engineering process in the automotive environment. It looks at the entire development process and life cycle of a vehicle following the V-model.

A larger collaboration between academic research and the practical implementation of security measures related to CAVs within the industry is required to determine attack vectors that can threaten the security, privacy, and safety of vehicle passengers and pedestrians. Table 39 describes the attacks and their potential countermeasures related to CAV subsystems, i.e., OBD, ECUs, CAN, LiDAR, Radar, GPS, image sensors, communication mechanisms, RSU, and VANET. However, understanding adversaries' motivation and capabilities (refer to Figure 9) for designing a water-tight security solution yet can be a critical challenge.



3 Less targeted attacks but capable attackers Organized Crime Malware	4 Targeted attacks and capable attackers State Sponsored Industrial Espionage
1 Less targeted attacks and Less capable attackers Prankster script kiddies	2 Targeted attacks but Less capable attackers Black hat hackers/evil doers Ethical hackers, Hacktivists Knowledgeable insiders

FIGURE 9. Adversaries motivation and capabilities [203]. *Quadrant 1* presents less targeted attacks by less capable attackers like pranksters and script kiddies, *Quadrant 2* presents targeted attacks for terrorism, hacktivism, or inside leaks by less capable adversaries. *Quadrant 3* focuses on attacks by capable adversaries to disrupt a system, and *Quadrant 4* presents the most serious attacks typically, with political or economic motives sponsored by a state or revelry industrial partner.

C. POTENTIAL RESEARCH AREAS

Researchers and engineers can evolve formal verification methodology for evaluating security mechanisms that can be employed for CAVs security. Furthermore, threat profiling can exploit the raw data collected from CAVs for

TABLE 39. CAV subsystems: attacks and countermeasures.

CAV Subsystems	References	Attacks	Countermeasures	Refer Section
OBD	[13], [7]	MiTM attack due to lack of data encryption or secure control access, malicious self-diagnostic applications (refer Table 12), DoS attack (refer Table 14)	Maintain authenticity, integrity, privacy	V-B1, V-B3
ECUs	[13], [15]	Code injection and reprogramming ECUs, power (refer Table 11) and electromagnetic side-channel attack (refer Table 19)	Secure firmware updates, robust access mechanism	V-A8, V-C2
CAN	[14], [13], [7], [15]	Eavesdropping, DoS attack, replay attack and unauthorized data transmission, bus fuzzing/frame falsifying/injection attack (refer Table 13, 14, 5, 11)	Confidentiality/authenticity/privacy of CAN messages, Intrusion Detection System (IDS) based defense techniques, securing telematics ECUs, cryptography, clock skew estimation of CAN messages, network segmentation	V-B2, V-B3, V-A2, V-A8
LiDAR	[13], [202], [72]	Spoofing, jamming, random modulation, signals counterfeiting (refer Table 13, 20)	Dynamic watermarking, reduction of signal-receiving angle, pulses transmission in random directions, and pulses' waveforms randomization, device fingerprinting, data authentication	V-B2, V-C3
Radar	[13], [202]	Spoofing, jamming, signals counterfeiting (refer Table 13, 20)	Signal filtering, Spatio-Temporal Challenge-Response (STCR)	V-B2, V-C3
GPS	[13], [14], [72]	Location trailing, spoofing, signals counterfeiting (refer Table 13, 20, 4)	Receiver autonomous integrity monitoring (RAIM), receiver-autonomous angle-of-arrival spoofing countermeasure, signal filtering	V-B2, V-C3, V-A1
Image sensors	[13], [202], [139], [72]	Camera blinding, adversarial images injection (refer Table 19, 11)	Deploy multiple cameras (redundancy), near-infrared-cut filter, special optics	V-C2, V-A8
Ultrasonic sensors	[72]	Spoofing, jamming (refer Table 13)	Acoustic quieting, e.g., cloaking and acoustic cancellation	V-B2
Communication mechanisms	[13], [7], [202]	Falsified information, DoS/DDoS, impersonation or Sybil attacks, eavesdropping (refer Table 14, 4, 13)	Robust authentication methods, neighboring nodes grouping, periodic Communication, IDS, consistency check	V-B3, V-A1, V-B2
Roadside unit (RSU)	[148], [15]	Timing attack, spoofing, DoS attack, Sybil attack (refer Table 36, 23, 14, 4)	Robust routing protocol, dynamic certificate scheme-based protection	V-C3, V-A2, V-B3, V-A1
Vehicular Ad hoc Networks (VANET)	[106], [147], [130]	Message alteration, DoS, impersonation attack, time and location-based attack refer Table 5, 14, 4, 36)	Trust management schemes, intrusion detection systems, Cryptography schemes	V-A2, V-B3, V-A1, V-C3, V-C4

generating more concrete information using AI that can be used by strategic applications to determine inherent weaknesses, identify possible threats, and predict zero-day cyber-attack scenarios that can adversely impact CAVs operations.

1) FORMAL VERIFICATION

With already a widespread application in designing safety-critical systems, formal verification is set to play an even more significant role in CAVs security. Studies have reported that formal verification (FV) can find weaknesses and possible vulnerabilities at the design stage [204]. Formal security verification can be highly effective in verifying data integrity and data leakage by tracing information flow. Formal verification of security mechanisms to be employed in CAV can mathematically prove the correctness of the underlying algorithms and software to accomplish the below objectives.

1) *Increased Safety*: An FV of the communication and control systems can reduce the likelihood of safety incidents. Passerone et al. [205] propose a contract-based approach for specifying safety, and augment it in the design flow using the Arrowhead Framework to support security. The proposed approach can address issues related to authentication and authorization of inter-vehicular signals and services carrying safety commands to ensure secure communication among vehicles.
2) *Improved Efficiency*: FV can help optimize the performance of security mechanisms for CAVs. Hofer-Schmitz and Stojanovic [204] describe that FV enables the correctness of designs by using a diverse set of mathematical and logical methods, thus, different parts of the system can be validated as the functional correctness of implementations and programming bugs, side-channel analysis, the fulfillment of security properties and hardware Trojans, right from designing phase and provide guarantees of security.
3) *Enhanced Reliability*: With FV, it is possible to detect and correct any errors in the design and implementation of the architecture, enhancing the system reliability. For instance, security properties such as confidentiality (secrecy) and authentication (authenticity of communicating parties) are essential for reliable communication protocols. The Dolev-Yao model has been widely used in the analysis and verification of cryptographic protocols and provides a formal framework to evaluate protocol security and identify potential vulnerabilities or attacks [206].
4) *Better Compliance*: a CAV architecture that undergoes FV is more likely to meet regulatory and industry standards, ensuring better compliance. It is recommended to introduce formal verification methodologies in ISO/SAE 21434 for strengthening the cybersecurity framework.
5) *Reduced Development Time*: Studies describe that using model-based approaches in which the models are rigorously specified enables the development of precise statements about what systems under investigation should do without putting constraints on how to do

it [206]. By verifying the security mechanisms through formal methods, it is possible to identify and correct design issues early on, reducing development time and cost.

6) *Better Quality Control*: FV can provide a more rigorous approach to quality control of (a) *safety-critical functions*, e.g., collision avoidance, emergency braking, or lane keeping, (b) *system-level behavior*, e.g., traffic rules adherence, safe distance maintenance from other vehicles, or correctly responding to various scenarios, (c) *sensor and perception systems*, e.g., correctness of sensor data processing algorithms, sensor fusion techniques, and object detection algorithms, and (d) *control algorithms*, e.g., vehicle motion and trajectory planning. Thus, improving the quality of the CAV ecosystem to operate in complex and unpredictable environments.
7) *Improved Interoperability*: CAVs' ability to communicate and interact seamlessly with each other is critical when they come from different manufacturers or have different levels of autonomy. FV techniques can be applied for various aspects of interoperability, such as (a) communication protocols to validate message-exchange between vehicles are properly formatted, interpreted, and understood for reducing the risk of miscommunication or misinterpretation, (b) safety-Critical Algorithms to verify the correctness and safety for complex algorithms for perception, decision-making, and control to avoid any hazards, (c) system Integration to validate correct functioning of sensors, actuators, and control systems, as well as their interaction with the communication protocols, and (d) fault tolerance and resilience to failures and faults for addressing unexpected situations or system failures during real-time operations. Thus, FV can ensure that the systems are interoperable, allowing for better integration with other technologies.

2) THREAT PROFILING

Given the complexity involved in the deployment of CAVs, zero-tolerance safety, and security measures must be incorporated to mitigate risks such as vehicle immobilization, road accidents, and disclosure of sensitive data [47], [72]. Existing security solutions can address known attacks or their subsets but to prevent complex attacks more deterministic real-time solutions are required. It is recommended to build real-time threat profile generation techniques that can facilitate deeper analysis opportunities and greater transparency to deal with systems uncertainly (*e.g., denial-of-service, black-box decisions, local discrepancy*), and common security problems (*e.g., social-engineering attacks, insider attacks, and sensitive data leakage*) [207].

VII. CONCLUSION

This article has taken take a comprehensive approach to studying cyber attacks on a typical CAVs ecosystem, i.e., CAVs as Edge devices, RSUs as Fog, and cloud servers as the backbone infrastructure. We have presented a common attack taxonomy derived by analyzing a reference architecture consisting of three sub-architecture for each tier: *Cloud*, *Edge*, and *CAVs* in the CAVs ecosystem. The taxonomy classifies existing and emerging cyber-attacks into software, network, and hardware that are comprehensively investigated for addressing the security requirements of the CAVs ecosystem. Subsequently, a detailed impact of twenty-one cyber-attacks on various systems and components is discussed for a thorough assessment of the CAVs ecosystem security. We have discussed potential security mechanisms that can be employed for protecting the hardware, network, and software components of the CAVs ecosystem to ensure safe and secure transportation.

A larger collaboration between academic research and the practical implementation of security measures addressing the security, privacy, and safety of vehicle passengers and pedestrians is highly crucial. Our studies have analyzed how each of the twenty-one attacks can affect the integral security properties, i.e., availability, authentication, integrity, confidentiality, non-repudiation, and privacy, and have investigated the impacts and outcomes of each attack on the entire CAVs ecosystem. We have also discussed the challenges and potential research areas, e.g., formal verification and threat profiling, that can provide insights to security engineers and system architects to design and develop a secure CAVs ecosystem.

REFERENCES

[1] Statista. (2023). *Size of the Global Connected Car Market Between 2019 and 2020, with a Forecast Through 2028*. [Online]. Available: https://www.statista.com/statistics/725025/connected-cars-global-market-size-projection/

[2] H. Leenstra, J. van den Berg, and B. van Wee, "Multi actor roadmap to improve cyber security of consumer used connected cars," Cyber Secur. Acad., Leiden Univ., Leiden, The Netherlands, Tech. Rep. S1727834, 2017.

[3] M. Endler, A. Silva, and R. A. M. S. Cruz, "An approach for secure edge computing in the Internet of Things," in *Proc. 1st Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2017, pp. 1–8.

[4] L. F. A. Leon, "Eyes on the road: Surveillance logics in the autonomous vehicle economy," *Surveill. Soc.*, vol. 17, nos. 1–2, pp. 198–204, Mar. 2019.

[5] Z. Ju, H. Zhang, X. Li, X. Chen, J. Han, and M. Yang, "A survey on attack detection and resilience for connected and automated vehicles: From vehicle dynamics and control perspective," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 4, pp. 815–837, Dec. 2022.

[6] T. Limbasiya, K. Z. Teng, S. Chattopadhyay, and J. Zhou, "A systematic survey of attack detection and prevention in connected and autonomous vehicles," *Veh. Commun.*, vol. 37, Oct. 2022, Art. no. 100515.

[7] E. Aliwa, O. Rana, C. Perera, and P. Burnap, "Cyberattacks and countermeasures for in-vehicle networks," *ACM Comput. Surv. (CSUR)*, vol. 54, no. 1, pp. 1–37, Jan. 2022.

[8] Ford. (2023). *Ford Safety Report: A Matter of 2.0*. [Online]. Available: https://media.ford.com/content/dam/fordmedia/North%20America/U.S./2021/06/17/ford-safety-report.pdf

[9] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, vol. 26, no. 3, pp. 1145–1168, Jun. 2021.

[10] S. Gupta, "Non-functional requirements elicitation for edge computing," *Internet Things*, vol. 18, May 2022, Art. no. 100503.

[11] S. Heinrich and L. Motors, "Flash memory in the emerging age of autonomy," in *Proc. Flash Memory Summit*, 2017, pp. 1–10.

[12] N. S. Vanitha, K. Radhika, M. Maheshwari, P. Suresh, and T. Meenakshi, "IoT-based intelligent transportation system for safety," in *Cloud and IoT-Based Vehicular Ad Hoc Networks*. Hoboken, NJ, USA: Wiley, 2021, pp. 47–65.

[13] M. Pham and K. Xiong, "A survey on security attacks and defense techniques for connected and autonomous vehicles," *Comput. Secur.*, vol. 109, Oct. 2021, Art. no. 102269.

[14] X. Sun, F. R. Yu, and P. Zhang, "A survey on cyber-security of connected and autonomous vehicles (CAVs)," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6240–6259, Jul. 2022.

[15] M. Dibaei, X. Zheng, K. Jiang, R. Abbas, S. Liu, Y. Zhang, Y. Xiang, and S. Yu, "Attacks and defences on intelligent connected vehicles: A survey," *Digit. Commun. Netw.*, vol. 6, no. 4, pp. 399–421, Nov. 2020.

[16] F. Sommer, J. Dürrwang, and R. Kriesten, "Survey and classification of automotive security attacks," *Information*, vol. 10, no. 4, p. 148, Apr. 2019.

[17] V. H. Le, J. den Hartog, and N. Zannone, "Security and privacy for innovative automotive applications: A survey," *Comput. Commun.*, vol. 132, pp. 17–41, Nov. 2018.

[18] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, "V2X access technologies: Regulation, research, and remaining challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1858–1877, 3rd Quart., 2018.

[19] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1243–1274, 2nd Quart., 2019.

[20] D. Sabella, H. Moustafa, P. Kuure, S. Kekki, Z. Zhou, A. Li, C. Thein, E. Fischer, I. Vukovic, J. Cardillo, V. Young, S. J. Tan, V. Park, M. Vanderveen, S. Runeson, and S. Sorrentino, "Toward fully connected vehicles: Edge computing for advanced automotive communications," 5G Automot. Assoc. (5GAA), Munich, Germany, White Paper no. 230, 2017.

[21] F. Arena and G. Pau, "When edge computing meets IoT systems: Analysis of case studies," *China Commun.*, vol. 17, no. 10, pp. 50–63, Oct. 2020.

[22] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.

[23] B. Fritzsche, "Revealing the invisible-information visualization in the Internet of Things era," in *Human Computer Interaction in the Internet of Things Era*, vol. 18. Munich, Germany: Univ. of Munich, 2015.

[24] J. Chang. (2017). *An Overview of USDOT Connected Vehicle Roadside Unit Research Activities*. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/34763

[25] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT edge computing with lightweight virtualization," *IEEE Netw.*, vol. 32, no. 1, pp. 102–111, Jan. 2018.

[26] S. Taherizadeh, A. C. Jones, I. Taylor, Z. Zhao, and V. Stankovski, "Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review," *J. Syst. Softw.*, vol. 136, pp. 19–38, Feb. 2018.

[27] 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, Nov. 2019.

[28] N. Huq, C. Gibson, and R. Vosseler. (2020). *Driving Security into Connected Cars: Threat Model and Recommendations*. [Online]. Available: https://documents.trendmicro.com/assets/white_papers/wp-driving-security-into-connected-cars.pdf

[29] E. Rask et al., "Smart mobility. Connected and automated vehicles capstone report," Lawrence Berkeley National Lab (LBNL), Berkeley, CA, USA, Tech. Rep., 2020.

[30] Edge Computing Consortium (ECC) and Alliance of Industrial Internet. (2017). *Edge Computing Reference Architecture 2.0*. [Online]. Available: http://en.ecconsortium.net/Uploads/file/20180328/1522232376480704.pdf

[31] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, 1st Quart., 2020.

[32] Y. Li, Q. Luo, J. Liu, H. Guo, and N. Kato, "TSP security in intelligent and connected vehicles: Challenges and solutions," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 125–131, Jun. 2019.

[33] Y. Shimizu, T. Kimura, and J. Cheng, "Detection method against fake message attacks in sparse mobile ad-hoc networks," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 1–7.

[34] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9762–9773, Dec. 2019.

[35] P. Sharma, D. Austin, and H. Liu, "Attacks on machine learning: Adversarial examples in connected and autonomous vehicles," in *Proc. IEEE Int. Symp. Technol. Homeland Secur. (HST)*, Nov. 2019, pp. 1–7.

[36] A. Sarker, H. Shen, T. Sen, and H. Uehara, "An advanced black-box adversarial attack for deep driving maneuver classification models," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Dec. 2020, pp. 184–192.

[37] W. Jiang, H. Li, S. Liu, X. Luo, and R. Lu, "Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4439–4449, Apr. 2020.

[38] C. Wang, J. Chen, Y. Yang, X. Ma, and J. Liu, "Poisoning attacks and countermeasures in intelligent networks: Status quo and prospects," *Digit. Commun. Netw.*, vol. 8, no. 2, pp. 225–234, Apr. 2022.

[39] N. Mashtalyar, U. N. Ntaganzwa, T. Santos, S. Hakak, and S. Ray, "Social engineering attacks: Recent advances and challenges," in *Proc. Int. Conf. Human-Comput. Interact.* Cham, Switzerland: Springer, 2021 pp. 417–431.

[40] A. Balueva, V. Desnitsky, and I. Ushakov, "Approach to detection of denial-of-sleep attacks in wireless sensor networks on the base of machine learning," in *Proc. Int. Symp. Intell. Distrib. Comput.* Cham, Switzerland: Springer, 2019, pp. 350–355.

[41] A. Gallais, T.-H. Hedli, V. Loscri, and N. Mitton, "Denial-of-sleep attacks against IoT networks," in *Proc. 6th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Apr. 2019, pp. 1025–1030.

[42] R. Smith, D. Palin, P. P. Ioulianou, V. G. Vassilakis, and S. F. Shahandashti, "Battery draining attacks against edge computing nodes in IoT networks," *Cyber-Phys. Syst.*, vol. 6, no. 2, pp. 96–116, Apr. 2020.

[43] V. Ponnusamy, N. D. Regunathan, P. Kumar, R. Annur, and K. Rafique, "A review of attacks and countermeasures in Internet of Things and cyber physical systems," in *Proc. Ind. Internet Things Cyber-Phys. Syst., Transforming Conventional Digit.* Philadelphia, PA, USA: IGI Global, 2020, pp. 1–24.

[44] G. E. Rodríguez, J. G. Torres, P. Flores, and D. E. Benavides, "Cross-Site Scripting (XSS) attacks and mitigation: A survey," *Comput. Netw.*, vol. 166, Jan. 2020, Art. no. 106960.

[45] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2027–2051, 3rd Quart., 2016.

[46] A. A. Ghali, R. Ahmad, and H. S. A. Alhussian, "Comparative analysis of DoS and DDoS attacks in Internet of Things environment," in *Proc. Comput. Sci. On-Line Conf.* Cham, Switzerland: Springer, 2020, pp. 183–194.

[47] S. Rizvi, J. Willet, D. Perino, S. Marasco, and C. Condo, "A threat to vehicular cyber security and the urgency for correction," *Proc. Comput. Sci.*, vol. 114, pp. 100–105, Oct. 2017.

[48] A. Jain and S. Jain, "A survey on miscellaneous attacks and countermeasures for RPL routing protocol in IoT," in *Proc. Emerg. Technol. Data Mining Inf. Secur.* Singapore: Springer, 2019, pp. 611–620.

[49] S. Gurung and S. Chauhan, "A survey of black-hole attack mitigation techniques in MANET: Merits, drawbacks, and suitability," *Wireless Netw.*, vol. 26, no. 3, pp. 1981–2011, Apr. 2020.

[50] K. M. A. Alheeti, A. Gruebler, and K. D. McDonald-Maier, "On the detection of grey hole and rushing attacks in self-driving vehicular networks," in *Proc. 7th Comput. Sci. Electron. Eng. Conf. (CEEC)*, Sep. 2015, pp. 231–236.

[51] A. Prabhakar and T. Anjali, "Gray hole attack as a Byzantine attack in a wireless multi-hop network," *J. Appl. Secur. Res.*, vol. 15, no. 1, pp. 116–145, Jan. 2020.

[52] S. Deshmukh-Bhosale and S. S. Sonavane, "A real-time intrusion detection system for wormhole attack in the RPL based Internet of Things," *Proc. Manuf.*, vol. 32, pp. 840–847, 2019.

[53] S. Agrawal, M. L. Das, and J. Lopez, "Detection of node capture attack in wireless sensor networks," *IEEE Syst. J.*, vol. 13, no. 1, pp. 238–247, Mar. 2019.

[54] S. Sidhu, B. J. Mohd, and T. Hayajneh, "Hardware security in IoT devices with emphasis on hardware Trojans," *J. Sensor Actuator Netw.*, vol. 8, no. 3, p. 42, Aug. 2019.

[55] S. Takarabt, A. Schaub, A. Facon, S. Guilley, L. Sauvage, Y. Souissi, and Y. Mathieu, ''Cache-timing attacks still threaten IoT devices,'' in *Proc. Int. Conf. Codes, Cryptol., Inf. Secur.* Cham, Switzerland: Springer, 2019, pp. 13–30.

[56] T. W. Edgar and D. O. Manz, ''Science and cyber security,'' in *Research Methods for Cyber Security*. Waltham, MA, USA: Syngress, 2017, ch. 2, pp. 33–62.

[57] W. Li, D. Mclernon, J. Lei, M. Ghogho, S. A. R. Zaidi, and H. Hui, ''Cryptographic primitives and design frameworks of physical layer encryption for wireless communications,'' *IEEE Access*, vol. 7, pp. 63660–63673, 2019.

[58] D. M. da Costa e Castro Mónica de, ''Thwarting The Sybil attack in wireless ad hoc networks,'' Jul. 2009. [Online]. Available: https://scholar.tecnico.ulisboa.pt/records/FzFiuOy8NwvQtpovKRsuc5yxE6u6-REyFucb

[59] A. Vasudeva and M. Sood, ''Survey on Sybil attack defense mechanisms in wireless ad hoc networks,'' *J. Netw. Comput. Appl.*, vol. 120, pp. 78–118, Oct. 2018.

[60] J. Li, Z. Xue, C. Li, and M. Liu, ''RTED-SD: A real-time edge detection scheme for Sybil DDoS in the Internet of Vehicles,'' *IEEE Access*, vol. 9, pp. 11296–11305, 2021.

[61] S. Chang, Y. Qi, H. Zhu, J. Zhao, and X. Shen, ''Footprint: Detecting Sybil attacks in urban vehicular networks,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 6, pp. 1103–1114, Jun. 2012.

[62] C. Wang, L. Zhu, L. Gong, Z. Zhao, L. Yang, Z. Liu, and X. Cheng, ''Accurate Sybil attack detection based on fine-grained physical channel information,'' *Sensors*, vol. 18, no. 3, p. 878, Mar. 2018.

[63] Y. Yao, B. Xiao, G. Yang, Y. Hu, L. Wang, and X. Zhou, ''Power control identification: A novel Sybil attack detection scheme in VANETs using RSSI,'' *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2588–2602, Nov. 2019.

[64] W. Zhang and G. Li, ''An efficient and secure data transmission mechanism for Internet of Vehicles considering privacy protection in fog computing environment,'' *IEEE Access*, vol. 8, pp. 64461–64474, 2020.

[65] K. Zhang, X. Liang, R. Lu, and X. Shen, ''Sybil attacks and their defenses in the Internet of Things,'' *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, Oct. 2014.

[66] D. S. Reddy, V. Bapuji, A. Govardhan, and S. S. V. N. Sarma, ''Sybil attack detection technique using session key certificate in vehicular ad hoc networks,'' in *Proc. Int. Conf. Algorithms, Methodol., Models Appl. Emerg. Technol. (ICAMMAET)*, Feb. 2017, pp. 1–5.

[67] N. Z. Gong, M. Frank, and P. Mittal, ''SybilBelief: A semi-supervised learning approach for structure-based Sybil detection,'' *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 6, pp. 976–987, Jun. 2014.

[68] Z. Yang, K. Zhang, L. Lei, and K. Zheng, ''A novel classifier exploiting mobility behaviors for Sybil detection in connected vehicle systems,'' *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2626–2636, Apr. 2019.

[69] J. J. Q. Yu, ''Sybil attack identification for crowdsourced navigation: A self-supervised deep learning approach,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4622–4634, Jul. 2021.

[70] Y. Lu, C. Maple, T. Sheik, H. Alhagagi, T. Watson, M. Dianati, and A. Mouzakitis, ''Analysis of cyber risk and associated concentration of research (ACR)^2 in the security of vehicular edge clouds,'' in *Proc. Living Internet Things, Cybersecurity IoT*, Mar. 2018, pp. 1–11.

[71] J. Raiyn, ''Data and cyber security in autonomous vehicle networks,'' *Transp. Telecommun. J.*, vol. 19, no. 4, pp. 325–334, Dec. 2018.

[72] 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.

[73] 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.

[74] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, ''Cybersecurity for autonomous vehicles: Review of attacks and defense,'' *Comput. Secur.*, vol. 103, Apr. 2021, Art. no. 102150.

[75] R.-M. Shi, Y.-B. Zhang, Y.-D. Zhang, Q.-R. Du, X.-B. Zhou, and X.-Z. Xu, ''A formal method for verifying the ability of a protocol to resist replay attacks,'' in *Proc. Int. Symp. Softw. Rel., Ind. Safety, Cyber Secur. Phys. Protection Nucl. Power Plant*. Singapore: Springer, 2019, pp. 238–247.

[76] W. Dai, C. Wang, C. Cui, H. Jin, and X. Lv, ''Blockchain-based smart contract access control system,'' in *Proc. 25th Asia–Pacific Conf. Commun. (APCC)*, Nov. 2019, pp. 19–23.

[77] K. Greene, D. Rodgers, H. Dykhuizen, K. McNeil, Q. Niyaz, and K. A. Shamaileh, ''Timestamp-based defense mechanism against replay attack in remote keyless entry systems,'' in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2020, pp. 1–4.

[78] V. Marquis, R. Ho, W. Rainey, M. Kimpel, J. Ghiorzi, W. Cricchi, and N. Bezzo, ''Toward attack-resilient state estimation and control of autonomous cyber-physical systems,'' in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2018, pp. 70–75.

[79] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, and X. Cheng, ''A distributed anomaly detection system for in-vehicle network using HTM,'' *IEEE Access*, vol. 6, pp. 9091–9098, 2018.

[80] D. K. Hong, J. Kloosterman, Y. Jin, Y. Cao, Q. A. Chen, S. Mahlke, and Z. M. Mao, ''AVGuardian: Detecting and mitigating publish-subscribe overprivilege for autonomous vehicle systems,'' in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Sep. 2020, pp. 445–459.

[81] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, ''Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4507–4518, Jul. 2021.

[82] J. Rauber, R. Zimmermann, M. Bethge, and W. Brendel, ''Foolbox Native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX,'' *J. Open Source Softw.*, vol. 5, no. 53, p. 2607, Sep. 2020.

[83] X. Gao, J. Liu, Y. Li, X. Wang, Y. Xiang, E. Tong, W. Niu, and Z. Han, ''Queue length estimation based defence against data poisoning attack for traffic signal control,'' in *Proc. Int. Conf. Intell. Inf. Process.* Cham, Switzerland: Springer, 2020, pp. 254–265.

[84] X. Wang, J. Li, X. Kuang, Y.-A. Tan, and J. Li, ''The security of machine learning in an adversarial setting: A survey,'' *J. Parallel Distrib. Comput.*, vol. 130, pp. 12–23, Aug. 2019.

[85] P. Xiong, S. Buffett, S. Iqbal, P. Lamontagne, M. Mamun, and H. Molyneaux, ''Towards a robust and trustworthy machine learning system development: An engineering perspective,'' *J. Inf. Secur. Appl.*, vol. 65, Mar. 2022, Art. no. 103121.

[86] S. Seetharaman, S. Malaviya, R. Vasu, M. Shukla, and S. Lodha, ''Influence based defense against data poisoning attacks in online learning,'' in *Proc. 14th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2022, pp. 1–6.

[87] P. P. K. Chan, Z.-M. He, H. Li, and C.-C. Hsu, ''Data sanitization against adversarial label contamination based on data complexity,'' *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 6, pp. 1039–1052, Jun. 2018.

[88] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, ''Casting out demons: Sanitizing training data for anomaly sensors,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2008, pp. 81–95.

[89] K. Sadeghi, A. Banerjee, and S. K. S. Gupta, ''A system-driven taxonomy of attacks and defenses in adversarial machine learning,'' *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 4, pp. 450–467, Aug. 2020.

[90] D. J. Miller, Z. Xiang, and G. Kesidis, ''Adversarial learning targeting deep neural network classification: A comprehensive review of defenses against attacks,'' *Proc. IEEE*, vol. 108, no. 3, pp. 402–433, Mar. 2020.

[91] R. A. Khamis, M. O. Shafiq, and A. Matrawy, ''Investigating resistance of deep learning-based IDS against adversaries using min-max optimization,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[92] P. Zhao, H. Jiang, J. Li, Z. Xiao, D. Liu, J. Ren, and D. Guo, ''Garbage in, garbage out: Poisoning attacks disguised with plausible mobility in data aggregation,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2679–2693, Jul. 2021.

[93] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, ''Evasion attacks against machine learning at test time,'' in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2013, pp. 387–402.

[94] F. Zhang, P. P. K. Chan, B. Biggio, D. S. Yeung, and F. Roli, ''Adversarial feature selection against evasion attacks,'' *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 766–777, Mar. 2016.

[95] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, and X. Yin, ''Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors,'' *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2632–2647, Aug. 2021.

[96] A. N. Bhagoji, D. Cullina, C. Sitawarin, and P. Mittal, "Enhancing robustness of machine learning systems via data transformations," in *Proc. 52nd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2018, pp. 1–5.

[97] R. Heartfield and G. Loukas, "Detecting semantic social engineering attacks with the weakest link: Implementation and empirical evaluation of a human-as-a-security-sensor framework," *Comput. Secur.*, vol. 76, pp. 101–127, Jul. 2018.

[98] X. Liang and Y. Kim, "A survey on security attacks and solutions in the IoT network," in *Proc. IEEE 11th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2021, pp. 0853–0859.

[99] H. Aldawood and G. Skinner, "Educating and raising awareness on cyber security social engineering: A literature review," in *Proc. IEEE Int. Conf. Teaching, Assessment, Learn. Eng. (TALE)*, Dec. 2018, pp. 62–68.

[100] W. Fan, K. Lwakatare, and R. Rong, "Social engineering: I-E based model of human weakness for attack and defense investigations," *Int. J. Comput. Netw. Inf. Secur.*, vol. 9, no. 1, pp. 1–11, Jan. 2017.

[101] T. Lauser, D. Zelle, and C. Krauß, "Security analysis of automotive protocols," in *Proc. Comput. Sci. Cars Symp.*, Dec. 2020, pp. 1–12.

[102] Y. Cui, X. Meng, Q. Chen, Y. Gao, C. Xu, S. Roberts, and Y. Wang, "Feasibility analysis of low-cost GNSS receivers for achieving required positioning performance in CAV applications," in *Proc. Forum Cooperat. Positioning Service (CPGPS)*, May 2017, pp. 355–361.

[103] A. S. Thangarajan, M. Ammar, B. Crispo, and D. Hughes, "Towards bridging the gap between modern and legacy automotive ECUs: A software-based security framework for legacy ECUs," in *Proc. IEEE 2nd Connected Automated Vehicles Symp. (CAVS)*, Sep. 2019, pp. 1–5.

[104] A. Yastrebova, T. Ojanperä, J. Mäkelä, and M. Höyhtyä, "Hybrid connectivity for autonomous vehicles: Conceptual view & initial results," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–6.

[105] S.-F. Lokman, A. T. Othman, and M.-H. Abu-Bakar, "Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–17, Dec. 2019.

[106] A. K. Malhi, S. Batra, and H. S. Pannu, "Security of vehicular ad-hoc networks: A comprehensive survey," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101664.

[107] S. Naik and N. Shekokar, "Conservation of energy in wireless sensor network by preventing denial of sleep attack," *Proc. Comput. Sci.*, vol. 45, pp. 370–379, Dec. 2015.

[108] E. Udoh and V. Getov, "Performance analysis of denial-of-sleep attack-prone MAC protocols in wireless sensor networks," in *Proc. UKSim-AMSS 20th Int. Conf. Comput. Model. Simul. (UKSim)*, Mar. 2018, pp. 151–156.

[109] T. Bhattasali and R. Chaki, "AMC model for denial of sleep attack detection," *J. Recent Res. Trends*, pp. 1–4, Oct. 2012.

[110] V. Desnitsky, N. Rudavin, and I. Kotenko, "Modeling and evaluation of battery depletion attacks on unmanned aerial vehicles in crisis management systems," in *Proc. Int. Symp. Intell. Distrib. Comput.* Cham, Switzerland: Springer, 2019, pp. 323–332.

[111] C.-T. Hsueh, C.-Y. Wen, and Y.-C. Ouyang, "A secure scheme against power exhausting attacks in hierarchical wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 6, pp. 3590–3602, Jun. 2015.

[112] V. C. Manju, S. L. S. Lekha, and M. S. Kumar, "Mechanisms for detecting and preventing denial of sleep attacks on wireless sensor networks," in *Proc. IEEE Conf. Inf. Commun. Technol.*, Apr. 2013, pp. 74–77.

[113] D. Airehrour, J. Gutierrez, and S. K. Ray, "Securing RPL routing protocol from blackhole attacks using a trust-based mechanism," in *Proc. 26th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Dec. 2016, pp. 115–120.

[114] 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)*, Dec. 2016, pp. 164–170.

[115] Y. Wang, N. Masoud, and A. Khojandi, "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1411–1421, Mar. 2021.

[116] Z. Cui, L. Du, P. Wang, X. Cai, and W. Zhang, "Malicious code detection based on CNNs and multi-objective algorithm," *J. Parallel Distrib. Comput.*, vol. 129, pp. 50–58, Jul. 2019.

[117] F. van Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1264–1276, Mar. 2020.

[118] S. Park and J.-Y. Choi, "Malware detection in self-driving vehicles using machine learning algorithms," *J. Adv. Transp.*, vol. 2020, Jan. 2020, Art. no. 3035741.

[119] N. V. Abhishek, M. N. Aman, T. J. Lim, and B. Sikdar, "DRiVe: Detecting malicious roadside units in the Internet of Vehicles with low latency data integrity," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3270–3281, Mar. 2022.

[120] D. Wei and X. Qiu, "Status-based detection of malicious code in Internet of Things (IoT) devices," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, May 2018, pp. 1–7.

[121] C. Zhao, J. S. Gill, P. Pisu, and G. Comert, "Detection of false data injection attack in connected and automated vehicles via cloud-based sandboxing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9078–9088, Jul. 2022.

[122] G. Barbu, L. Bettale, L. Castelnovi, T. Chabrier, N. Debande, C. Giraud, and N. Reboud, "A high-order infective countermeasure framework," in *Proc. Workshop Fault Detection Tolerance Cryptogr. (FDTC)*, Sep. 2021, pp. 13–19.

[123] D. Cotroneo, L. De Simone, and R. Natella, "Run-time detection of protocol bugs in storage I/O device drivers," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 847–869, Sep. 2018.

[124] D. Mitropoulos, P. Louridas, M. Polychronakis, and A. D. Keromytis, "Defending against web application attacks: Approaches, challenges and implications," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 2, pp. 188–203, Mar. 2019.

[125] L. Xue, Y. Liu, T. Li, K. Zhao, J. Li, X. L. Le Yu, Y. Zhou, and G. Gu, "SAID: State-aware defense against injection attacks on in-vehicle network," in *Proc. 31st USENIX Secur. Symp.* Boston, MA, USA: USENIX Association, 2022, pp. 1–18.

[126] S. Jeong, B. Jeon, B. Chung, and H. K. Kim, "Convolutional neural network-based intrusion detection system for AVTP streams in automotive Ethernet-based networks," *Veh. Commun.*, vol. 29, Jun. 2021, Art. no. 100338.

[127] A. Birnie and T. van Roermund, "A multi-layer vehicle security framework," NXP Semicond., White Paper, May 2016. [Online]. Available: https://www.nxp.com/docs/en/white-paper/MULTI-LAYER-VEHICLE-SECURITY-WP.pdf

[128] N. S. Nambiar, C. Tubakad, A. Kiran, and S. Kalambur, "Multilevel secure container deployment framework in edge computing," in *Proc. Int. Symp. Secur. Comput. Commun.* Singapore: Springer, 2020 pp. 49–61.

[129] S. Jasek. (2015). *Connected Car Security Threat Analysis and Recommendations*. [Online]. Available: https://www.securing.pl/en/connected-car-security-threat-analysis-and-recommendations/

[130] M. Azees, P. Vijayakumar, and L. J. Deborah, "Comprehensive survey on security services in vehicular ad-hoc networks," *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 379–388, 2016.

[131] F. Aliyu, T. Sheltami, and E. M. Shakshuki, "A detection and prevention technique for man in the middle attack in fog computing," *Proc. Comput. Sci.*, vol. 141, pp. 24–31, 2018.

[132] W. Yang, X. Li, Z. Feng, and J. Hao, "TLSsem: A TLS security-enhanced mechanism against MITM attacks in public WiFis," in *Proc. 22nd Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Nov. 2017, pp. 30–39.

[133] W. Shen, Y. Cheng, B. Yin, J. Du, and X. Cao, "Diffie–Hellman in the air: A link layer approach for in-band wireless pairing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11894–11907, Nov. 2021.

[134] A. Chhabra and S. Arora, "An elliptic curve cryptography based encryption scheme for securing the cloud against eavesdropping attacks," in *Proc. IEEE 3rd Int. Conf. Collaboration Internet Comput. (CIC)*, Oct. 2017, pp. 243–246.

[135] S. Choi, S. Han, and J.-W. Choi, "A secure transmission scheme at the receiver for eavesdropping prevention," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.

[136] B. Wu and C. Wang, "A privacy preserving network coding signature scheme based on lattice," *J. Phys., Conf. Ser.*, vol. 1693, no. 1, Dec. 2020, Art. no. 012048.

[137] Z. Li, Y. Zhu, and K. G. Shin, "iCoding: Countermeasure against interference and eavesdropping in wireless communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[138] C.-H. Liao, H.-H. Shuai, and L.-C. Wang, "Eavesdropping prevention for heterogeneous Internet of Things systems," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–2.

[139] L. Cao, X. Jiang, Y. Zhao, S. Wang, D. You, and X. Xu, "A survey of network attacks on cyber-physical systems," *IEEE Access*, vol. 8, pp. 44219–44227, 2020.

[140] G. R. Andreica, L. Bozga, D. Zinca, and V. Dobrota, "Denial of service and man-in-the-middle attacks against IoT devices in a GPS-based monitoring software for intelligent transportation systems," in *Proc. 19th RoEduNet Conf., Netw. Educ. Res. (RoEduNet)*, Dec. 2020, pp. 1–4.

[141] R. Nanda and P. V. Krishna, "Mitigating denial of service attacks in hierarchical wireless sensor networks," *Netw. Secur.*, vol. 2011, no. 10, pp. 14–18, Oct. 2011.

[142] C. Lyu, D. Gu, Y. Zeng, and P. Mohapatra, "PBA: Prediction-based authentication for vehicle-to-vehicle communications," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 1, pp. 71–83, Jan. 2016.

[143] Z. Tian, X. Gao, S. Su, and J. Qiu, "Vcash: A novel reputation framework for identifying denial of traffic service in Internet of Connected Vehicles," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3901–3909, May 2020.

[144] X. Wang, J. H. Park, H. Liu, and X. Zhang, "Cooperative output-feedback secure control of distributed linear cyber-physical systems resist intermittent DoS attacks," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4924–4933, Oct. 2021.

[145] D. Zhang, Y.-P. Shen, S.-Q. Zhou, X.-W. Dong, and L. Yu, "Distributed secure platoon control of connected vehicles subject to DoS attack: Theory and application," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 11, pp. 7269–7278, Nov. 2021.

[146] Z. A. Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3893–3902, Dec. 2018.

[147] C. Lai, K. Zhang, N. Cheng, H. Li, and X. Shen, "SIRC: A secure incentive scheme for reliable cooperative downloading in highway VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1559–1574, Jun. 2017.

[148] M. Kabbur and V. A. Kumar, "MAR_Sybil: Cooperative RSU based detection and prevention of Sybil attacks in routing process of VANET," *J. Phys., Conf. Ser.*, vol. 1427, no. 1, Jan. 2020, Art. no. 012009.

[149] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge-computing-assisted Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4004–4022, Mar. 2021.

[150] S. Nayak, N. Ahmed, and S. Misra, "Deep learning-based reliable routing attack detection mechanism for Industrial Internet of Things," *Ad Hoc Netw.*, vol. 123, Dec. 2021, Art. no. 102661.

[151] M. Wazid, P. Reshma Dsouza, A. K. Das, V. Bhat K, N. Kumar, and J. J. P. C. Rodrigues, "RAD-EI: A routing attack detection scheme for edge-based Internet of Things environment," *Int. J. Commun. Syst.*, vol. 32, no. 15, p. e4024, Oct. 2019.

[152] HeLiu, Y. Zhao, and Q. Dong, "Anomaly detection for DOS routing attack by a attack source location method," in *Proc. IEEE Chin. Guid., Navigat. Control Conf. (CGNCC)*, Aug. 2016, pp. 25–29.

[153] T. Shu and M. Krunz, "Privacy-preserving and truthful detection of packet dropping attacks in wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 4, pp. 813–828, Apr. 2015.

[154] R. M. G. Ferrari and A. M. H. Teixeira, "Detection and isolation of routing attacks through sensor watermarking," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 5436–5442.

[155] Y. Wang, Z. Qi, X. Sun, Z. Xiang, and Y. Chen, "Recent development of security issues of black hole and gray hole attacks in V2X network," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[156] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: Secure and trustable routing in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2013–2027, Sep. 2016.

[157] J. Tobin, C. Thorpe, and L. Murphy, "An approach to mitigate black hole attacks on vehicular wireless networks," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–7.

[158] N. Schweitzer, A. Stulman, R. D. Margalit, and A. Shabtai, "Contradiction based gray-hole attack minimization for ad-hoc networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2174–2183, Aug. 2017.

[159] J. Hua, Z. Zhou, and S. Zhong, "Flow misleading: Worm-hole attack in software-defined networking via building in-band covert channel," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1029–1043, 2021.

[160] S. Jagadeesan and V. Parthasarathy, "Design and implement a cross layer verification framework (CLVF) for detecting and preventing blackhole and wormhole attack in wireless ad-hoc networks for cloud environment," *Cluster Comput.*, vol. 22, no. 1, pp. 299–310, 2019.

[161] C. K. Doshi, S. Sankaranarayanan, V. B. Lakshman, and K. Chandrasekaran, "Game theoretic modeling of gray hole attacks in wireless ad hoc networks," in *Proc. Int. Conf. Signal, Netw., Comput., Syst.*, in Lecture Notes in Electrical Engineering, vol. 395. Springer, 2017, pp. 217–226. [Online]. Available: https://link.springer.com/chapter/10.1007/978-81-322-3592-7_21

[162] M. Conti, "Capture detection," in *Secure Wireless Sensor Networks*. Springer, 2016, pp. 53–73. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4939-3460-7_3

[163] W. Li and P. Wang, "Two-factor authentication in industrial Internet-of-things: Attacks, evaluation and new construction," *Future Gener. Comput. Syst.*, vol. 101, pp. 694–708, Dec. 2019.

[164] L. Li, G. Xu, L. Jiao, X. Li, H. Wang, J. Hu, H. Xian, W. Lian, and H. Gao, "A secure random key distribution scheme against node replication attacks in industrial wireless sensor systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2091–2101, Mar. 2020.

[165] J. Zhao, "On resilience and connectivity of secure wireless sensor networks under node capture attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 557–571, Mar. 2017.

[166] J. Gu, J. Wang, and H. Sun, "Detection of node capture under asynchronous sleep mode based on recurrent neural network," *Converter*, vol. 2021, pp. 198–208, Jul. 2021.

[167] Y. Zhang and P. Li, "Key management scheme based on nodes capture probability for wireless sensor networks," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 5470–5475.

[168] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 507–523, Jan. 2022.

[169] E. Poonguzhali, A. Priyadarsini, P. Magnifique, and S. Asvini, "A security model for timing attack in cloud environment," in *Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS)*, Mar. 2015, pp. 1–5.

[170] I. Pekaric, C. Sauerwein, S. Haselwanter, and M. Felderer, "A taxonomy of attack mechanisms in the automotive domain," *Comput. Standards Interfaces*, vol. 78, Oct. 2021, Art. no. 103539.

[171] P. Kiaei, D. Mercadier, P.-E. Dagand, K. Heydemann, and P. Schaumont, "Custom instruction support for modular defense against side-channel and fault attacks," in *Proc. Int. Workshop Constructive Side-Channel Anal. Secure Design*. Cham, Switzerland: Springer, 2020, pp. 221–253.

[172] J. Wang, Y. Yang, L. Chen, G. Yang, Z. Chen, and L. Wen, "A combination of timing attack and statistical method to reduce computational complexities of SSL/TLS side-channel attacks," in *Proc. 11th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2015, pp. 402–406.

[173] J. Park, M. Corba, A. E. de la Sema, R. L. Vigeant, M. Tehranipoor, and S. Bhunia, "ATAVE: A framework for automatic timing attack vulnerability evaluation," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 559–562.

[174] W. Liu, D. Gao, and M. K. Reiter, "On-demand time blurring to support side-channel defense," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, 2017, pp. 210–228.

[175] S. Chattopadhyay and A. Roychoudhury, "Symbolic verification of cache side-channel freedom," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2812–2823, Nov. 2018.

[176] J. Chen, Y. Feng, and I. Dillig, "Precise detection of side-channel vulnerabilities using quantitative Cartesian Hoare Logic," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 875–890.

[177] C. Lavaud, R. Gerzaguet, M. Gautier, O. Berder, E. Nogues, and S. Molton, "Whispering devices: A survey on how side-channels lead to compromised information," *J. Hardw. Syst. Secur.*, vol. 5, pp. 143–168, Mar. 2021.

[178] S. Benzarti, B. Triki, and O. Korbaa, "A survey on attacks in Internet of Things based networks," in *Proc. Int. Conf. Eng. MIS (ICEMIS)*, May 2017, pp. 1–7.

[179] L. Li, J. Sun, Y. Liu, M. Sun, and J.-S. Dong, "A formal specification and verification framework for timed security protocols," *IEEE Trans. Softw. Eng.*, vol. 44, no. 8, pp. 725–746, Aug. 2018.

[180] C. Li, Q. Han, B. Lei, H. Liu, C. Liu, and Y. He, "Timing attacks in single-chip microcomputer through workflow verification," in *Proc. 8th Int. Conf. Dependable Syst. Their Appl. (DSA)*, Aug. 2021, pp. 716–721.

[181] V. Selis and A. Marshall, "A fake timing attack against behavioural tests used in embedded IoT M2M communications," in *Proc. 1st Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2017, pp. 1–6.

[182] S. Peter and T. Givargis, "Towards a timing attack aware high-level synthesis of integrated circuits," in *Proc. IEEE 34th Int. Conf. Comput. Design (ICCD)*, Oct. 2016, pp. 452–455.

[183] T. Meng, K. Wolter, H. Wu, and Q. Wang, "A secure and cost-efficient offloading policy for mobile cloud computing against timing attacks," *Pervas. Mobile Comput.*, vol. 45, pp. 4–18, Apr. 2018.

[184] A. Javeed, C. Yilmaz, and E. Savas, "Detector$^+$: An approach for detecting, isolating, and preventing timing attacks," *Comput. Secur.*, vol. 110, Nov. 2021, Art. no. 102454.

[185] M. Devi and A. Majumder, "Side-channel attack in Internet of Things: A survey," in *Applications of Internet of Things*. Singapore: Springer, 2021, pp. 213–222.

[186] 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.

[187] M. M. Sravani and S. Ananiah Durai, "Attacks on cryptosystems implemented via VLSI: A review," *J. Inf. Secur. Appl.*, vol. 60, Aug. 2021, Art. no. 102861.

[188] M. Babu and G. A. S. Kumar, "Design of novel SMS4-BSK encryption transmission system," *Integration*, vol. 78, pp. 60–69, May 2021.

[189] W. Li, D. McLernon, J. Lei, M. Ghogho, S. A. R. Zaidi, and H. Hui, "Mathematical model and framework of physical layer encryption for wireless communications," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–7.

[190] R. D. Mushrall, M. D. Furtado, and H. Liu, "EmuLab of Security Credential Management System (SCMS) for vehicular communications," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5.

[191] S. Gupta, K. Raja, F. Martinelli, and B. Crispo, "RiderAuth: A cancelable touch-signature based rider authentication scheme for driverless taxis," *J. Inf. Secur. Appl.*, vol. 71, Dec. 2022, Art. no. 103357.

[192] AUTO-ISAC. (2023). *Automotive Information Sharing and Analysis Center*. [Online]. Available: https://automotiveisac.com/

[193] Waymo. (2023). *Waymo Safety Report*. [Online]. Available: https://storage.googleapis.com/waymo-uploads/files/documents/safety/2021-12-waymo-safety-report.pdf

[194] Toyota. (2023). *Toyota Acknowledges Tencent Keen Security Lab's Initiatives for Improving Automotive Cybersecurity*. [Online]. Available: https://global.toyota/en/newsroom/corporate/32120629.html

[195] Wayve. (2023). *Safety Framework*. [Online]. Available: https://wayve.ai/safety/safety-framework/

[196] NXP. (2023). *NXP and Ford Collaborate to Deliver Next-Generation Connected Car Experiences and Expanded Services*. [Online]. Available: https://www.nxp.com/company/about-nxp/nxp-and-ford-collaborate-to-deliver-next-generation-connected-car-experiences-and-expanded-services:nw-nxp-and-ford-collaborate-to-deliver-next-gen

[197] T. Cui, L. Li, Z. Zhang, and C. Sun, "C-V2X vision in the Chinese roadmap: Standardization, field tests, and industrialization," in *Vehicular Networks—Principles, Enabling Technologies and Perspectives*. London, U.K.: IntechOpen, 2022, ch. 3, pp. 1–6.

[198] Cavnue. (2023). *Building the Future of Roads*. [Online]. Available: https://www.cavnue.com/news/cavnue-narrative/

[199] (2023). *ISO/SAE 21434:2021(EN) Road Vehicles—Cybersecurity Engineering*. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-sae:21434:ed-1:v1:en

[200] E. Brenner, "ISO/SAE DIS 21434 automotive cybersecurity standard—In a nutshell," in *Proc. Comput. Saf., Rel., Secur.*, vol. 12235. Cham, Switzerland: Springer, 2020, pp. 123–135.

[201] A. Barber, "Status of work in process on ISO/SAE 21434 automotive cybersecurity standard," presented at the ISO SAE International, Warrendale, PA, USA, Apr. 2018.

[202] 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.

[203] S. Liebl, L. Lathrop, U. Raithel, M. Söllner, and A. Aßmuth, "Threat analysis of industrial Internet of Things devices," in *Proc. 11th Int. Conf. Cloud Comput.*, 2020, pp. 31–37.

[204] K. Hofer-Schmitz and B. Stojanović, "Towards formal verification of IoT protocols: A review," *Comput. Netw.*, vol. 174, Jun. 2020, Art. no. 107233.

[205] R. Passerone, D. Cancila, M. Albano, S. Mouelhi, S. Plosz, E. Jantunen, A. Ryabokon, E. Laarouchi, C. Hegedus, and P. Varga, "A methodology for the design of safety-compliant and secure communication of autonomous vehicles," *IEEE Access*, vol. 7, pp. 125022–125037, 2019.

[206] T. Kulik, B. Dongol, P. G. Larsen, H. D. Macedo, S. Schneider, P. W. V. Tran-Jørgensen, and J. Woodcock, "A survey of practical formal methods for security," *Formal Aspects Comput.*, vol. 34, no. 1, pp. 1–39, Mar. 2022.

[207] S. Gupta, "An edge-computing based Industrial Gateway for Industry 4.0 using ARM TrustZone technology," *J. Ind. Inf. Integr.*, vol. 33, Jun. 2023, Art. no. 100441.

SANDEEP GUPTA received the Ph.D. degree in information and communication technology from the University of Trento, Italy. He was a recipient of the Prestigious Marie Sklodowska-Curie Research Fellowship. He is currently with the University of Trento. From 1999 to 2016, he was with Samsung, Accenture, and Mentor Graphics (now Siemens) in information technology domains driving several software products and research projects from their incubation to next-generation iteration. He also founded Apache Technologies and co-founded FadFudge. Since 2016, he has been working on EU H2020 projects—Collabs, E-Corridor, NeCS, and CyberSec4Europe. He has published more than 25 papers in peer-reviewed journals and leading conferences. His research interests include trustworthy AI, biometric-based identity and access mechanisms, usable security and privacy solutions for cyber-physical systems, and the IoT.

CARSTEN MAPLE is currently the Principal Investigator of the NCSC-EPSRC Academic Centre of Excellence in Cyber Security Research, and a Professor of cyber systems engineering with the Warwick Manufacturing Group (WMG). He is also a Co-Investigator of the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, where he leads on transport and mobility. He is a fellow of the Alan Turing Institute, and the National Institute for Data Science and AI, U.K., where he is a principal investigator on a $5 million project developing trustworthy national identity to enable financial inclusion. He has an international research reputation and extensive experience of institutional strategy development and interacting with external agencies. He has published over 250 peer-reviewed papers and is the coauthor of the U.K. Security Breach Investigations Report 2010, supported by the Serious Organized Crime Agency and the Police Central E-Crime Unit. He is also the coauthor of Cyberstalking in the U.K., a report supported by the Crown Prosecution Service and Network for Surviving Stalking. His research has attracted millions of pounds in funding and he has been widely reported through the media. He has given evidence to government committees on a variety of issues concerning safety, security, privacy, and identity.

ROBERTO PASSERONE (Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, in 1997 and 2004, respectively. He is currently an Associate Professor in electronics with the Department of Information Engineering and Computer Science, University of Trento, Italy. Before joining the University of Trento, he was a Research Scientist with Cadence Design Systems. He has published numerous research articles on international conferences and journals in the area of design methods for systems and integrated circuits, formal models, and design methodologies for embedded systems, with particular attention to image processing, and wireless sensor networks. He has participated in several European projects on design methodologies, including SPEEDS, SPRINT, and DANSE, and he was the Local Coordinator for ArtistDesign, COMBEST, and CyPhERS. He has served as the Track Chair for the Realtime and Networked Embedded Systems with ETFA, from 2008 to 2010, and the General Chair and the Program Chair for various editions of SIES.

Open Access funding provided by 'Università degli Studi di Trento' within the CRUI CARE Agreement