Contextualizing Security and Privacy of Software-Defined Vehicles: A Literature Review and Industry Perspectives

MARCO DE VINCENZI, Institute for Informatics and Telematics, CNR, Italy
MERT D. PESÉ, School of Computing, Clemson University, USA
CHIARA BODEI, Department of Computer Science, Università di Pisa, Italy
ILARIA MATTEUCCI, Institute for Informatics and Telematics, CNR, Italy
RICHARD R. BROOKS, Department of Electrical and Computer Engineering, Clemson University, USA
MONOWAR HASAN, School of EECS, Washington State University, USA
ANDREA SARACINO, TeCIP Institute, Sant'Anna School of Advanced Studies, Italy
MOHAMMAD HAMAD, Technical University of Munich, Germany
SEBASTIAN STEINHORST, Technical University of Munich, Germany

The growing reliance on software in road vehicles has led to the emergence of Software-Defined Vehicles (SDV). This work analyzes SDV security and privacy through a systematic literature review complemented by an industry questionnaire across the automotive supply chain. The analysis is structured as four research questions and results in a security framework serving as a roadmap for SDV protection. The findings emphasize addressing mixed-criticality architectural challenges, deploying layered security mechanisms, and integrating privacy-preserving techniques. The results highlight the need to harmonize in-vehicle and cloud-based defenses to strengthen cybersecurity and V2X resilience in Intelligent Transportation Systems (ITS).

CCS Concepts: • **Security and privacy → Distributed systems security**.

Additional Key Words and Phrases: Software-Defined Vehicle (SDV), Security, Privacy, Over-the-Air (OTA), E/E Architecture.

ACM Reference Format:
Marco De Vincenzi, Mert D. Pesé, Chiara Bodei, Ilaria Matteucci, Richard R. Brooks, Monowar Hasan, Andrea Saracino, Mohammad Hamad, and Sebastian Steinhorst. 2026. Contextualizing Security and Privacy of Software-Defined Vehicles: A Literature Review and Industry Perspectives. *ACM Comput. Surv.* 1, 1, Article 1 (January 2026), 35 pages. https://doi.org/XXXXXXX.XXXXXXX

Authors' addresses: Marco De Vincenzi, marco.devincenzi@iit.cnr.it, Institute for Informatics and Telematics, CNR, Via Giuseppe Moruzzi 1, Pisa, Italy, 56124; Mert D. Pesé, mpese@clemson.edu, School of Computing, Clemson University, 215 McAdams Hall, Clemson, SC, USA, 29634; Chiara Bodei, chiara.bodei@unipi.it, Department of Computer Science, Università di Pisa, Largo Bruno Pontecorvo, 3, Pisa, Italy, 56127; Ilaria Matteucci, ilaria.matteucci@iit.cnr.it, Institute for Informatics and Telematics, CNR, Via Giuseppe Moruzzi 1, Pisa, Italy, 56124; Richard R. Brooks, rrb@clemson.edu, Department of Electrical and Computer Engineering, Clemson University, 313C Riggs Hall, Clemson, SC, USA, 29634; Monowar Hasan, monowar.hasan@wsu.edu, School of EECS, Washington State University, 355 NE Spokane St, Pullman, WA, USA, 99164-2920; Andrea Saracino, andrea.saracino@santannapisa.it, TeCIP Institute, Sant'Anna School of Advanced Studies, Piazza Martiri della Libertà 33, Pisa, Italy, 56127; Mohammad Hamad, mohammad.hamad@tum.de, Technical University of Munich, Arcisstr., 21, München, Germany, D-80333; Sebastian Steinhorst, sebastian.steinhorst@tum.de, Technical University of Munich, Arcisstr., 21, München, Germany, D-80333.

ACM 0360-0300/2026/1-ART1
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Software-Defined Vehicles (SDVs) represent an emerging trend in the automotive industry [41, 91, 147], where vehicle functionalities, behaviors, and features are increasingly controlled by software rather than hardware components [20, 71, 119]. It is estimated that by 2040, 40% of the total revenue of the automotive industry will come from digitally enabled services [2, 69]. Recent industry analyses indicate that the convergence of Artificial Intelligence (AI) solutions, such as Vision-Language Models (VLM), and software-defined architectures is transforming vehicles into adaptive, continuously evolving platforms, positioning SDVs as a cornerstone for next-generation automotive innovation and competitiveness [15, 123]. This paradigm shift suggests a gradual redesign of current architectures and the entire industry, providing users with benefits such as greater flexibility, easier upgrades, personalized customization, and the integration of autonomous technologies [138]. This evolution affects not only development and operations, but also enables new business models and forms of collaboration, such as partnerships between Original Equipment Manufacturers (OEMs) and technology companies [24]. In this context, firms traditionally rooted in the technology sector are increasingly entering the automotive domain and operating as vehicle manufacturers or mobility providers [32, 92].

The growing integration of software components has been accompanied by a rise in cyberattacks. This trend dates back to one of the first remote vehicle exploits, which targeted a Jeep Cherokee in 2014 [28]. It continued in subsequent years, including the discovery of 14 vulnerabilities in BMW vehicles by engineers from the Keen Security Lab in 2018 [144], and the identification of flaws in a Tesla Model 3 by a German teenager in 2022 [97]. More recently, in 2024, Sam Curry and his team demonstrated the ability to remotely control Kia vehicle functionality using only the license plate [34]. Looking ahead, AI is expected to become a key enabler for threat actors, allowing rapid vulnerability discovery and exploitation, and potentially supporting fleet-wide attacks [130]. Moreover, recent incidents involving modified communication devices that were remotely activated and controlled [82] have heightened concerns about the integrity of the production supply chain. Such attacks, particularly when vehicles are deliberately deployed or compromised, may transform vehicles into weapons, as observed in vehicle ramming incidents, posing severe risks to public safety [13, 35, 77, 145]. Finally, the increasing complexity and volume of vehicle software, together with the ongoing technological transformation of vehicles, have expanded the number of potential attack surfaces and vectors [26].

From an industrial perspective, the ISO/SAE 21434 standard [76] and the UNECE regulations R155/156 [148, 149] were introduced in 2021 to address cybersecurity vulnerabilities in Connected Vehicles (CVs) and SDVs. These security frameworks aim to harmonize and strengthen cybersecurity practices across the automotive industry. However, as the transition toward UNECE R155 compliance advances, open questions remain regarding the implementation of new cybersecurity and software update infrastructures. A key challenge concerns the effective application of "Security by Design" principles in emerging software architectures. As an example, the de facto standard AUTomotive Open System ARchitecture (AUTOSAR) [9] introduced the Adaptive Platform to address the security requirements in SDV architectures by adopting a dynamic, service-oriented, POSIX-based approach, in contrast to the Classic Platform tailored to hardware-centric vehicle architectures. This evolution is under active development as SDV requirements continue to mature [10]. Furthermore, additional challenges arise in coordinating cybersecurity with SOTIF (Safety of the Intended Functionality) and functional safety requirements defined by ISO 26262 [75].

This work investigates security and privacy issues in SDVs by analyzing insights from the literature and industry feedback on current challenges and potential solutions. As the concept of SDV [20] is still evolving and lacks a consolidated definition, foundational concepts relevant to

SDVs' security are first introduced. These include a definition building on prior work [20] and an analysis of the differences between SDVs, Autonomous Vehicles (AVs), and CVs. The study is conducted as a systematic literature review, structured around a foundational section and four Research Questions (RQs). Relevant literature is systematically collected and analyzed following the review methodology described in Section 2.1. To further strengthen the analysis, feedback from industry experts is incorporated through a targeted elicitation (Section 2.2). This integration addresses both the limited availability of literature on this emerging topic and the need to capture practical insights from professionals directly involved in SDV development. Their contributions on key aspects, security threats, and mitigation strategies complement the literature and provide a more comprehensive view of the current state of SDV development.

The results show that the transition to SDVs introduces challenges and risks in security and privacy, mainly due to the increased reliance on software and attack surfaces within vehicles. These risks range from API vulnerabilities and third-party software risks to complex supply chain threats and potential privacy infringements. Industry feedback further underscores the urgent need for robust, standardized security frameworks and privacy-preserving mechanisms to support the evolution of SDVs. In addition, the findings highlight the need to adopt multilayered security measures and to integrate in-vehicle and cloud-based solutions to protect future SDVs, while increasing user trust, particularly in the widespread adoption of AI-based solutions within Intelligent Transportation Systems (ITSs).

1.1 Organization of the Paper

This work is structured into three main sections: *(I)* a foundational section (Section 3) that presents a definition of SDVs to highlight their features, distinguish them as unique entities, and clarify the differences between SDVs, CVs, and AVs (Section 3.3); *(II)* a section (Section 4) addressing SDV security, which includes two research questions on attack surfaces (Section 4.1, **RQ1**) and mitigations (Section 4.2, **RQ2**); and *(III)* a section (Section 5) on SDV security challenges that examines two research questions focusing on Over-the-Air (OTA) (Section 5.1, **RQ3**) and SDV data privacy (Section 5.2, **RQ4**). These aspects are selected for their critical roles in the functionality and security of SDVs. OTA updates enable continuous improvement and maintenance of SDVs, while SDVs generate and process vast amounts of sensitive data. Figure 1 summarizes the software-centric ecosystem, the proposed RQs, and the main actors involved. It also illustrates the interconnections among SDV security challenges, showing how they inform the threat and mitigation sections. Each section concludes with a gray box highlighting the key takeaways. Finally, Section 7 presents the overall conclusions. The RQs are as follows:
RQ1 - What are the attack surfaces and associated threats in SDVs?
RQ2 - What strategies can mitigate attack surface vulnerabilities?
RQ3 - What are the main security issues that OTA updates face, including the outside vehicle environment?
RQ4 - How do SDVs affect data collection, and what are the primary concerns related to user and vehicle privacy?

1.2 Motivations and Contributions

Considering the rapid evolution of vehicles in recent years and the absence of a comprehensive definition and systematic analysis of SDV security and privacy, the motivations behind this work are fourfold, as outlined below.

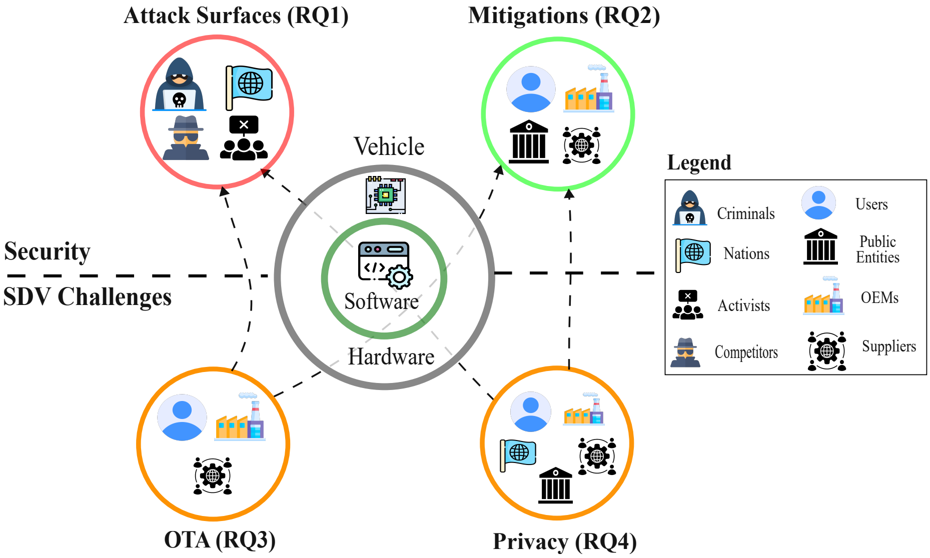


Fig. 1. The review structure with the new software-centric ecosystem, including the RQs.

Technological evolution: the implications of the transition from hardware-centric to software-centric vehicle systems need to be explored, as this shift is rapidly redefining the boundaries and capabilities of modern vehicles [20, 71, 119].

Security and privacy concerns: the growing concerns surrounding security and privacy risks associated with SDVs need to be examined, particularly in light of recent high-profile cyberattacks on CVs and in the entire CV ecosystem [34, 40, 97, 144].

Need for comprehensive analysis: in view of the requirements set by UNECE regulations R155/156 [148, 149] and the ISO/SAE 21434 standard [76], a detailed review of existing literature and current practices is provided to identify knowledge gaps and potential areas for compliance and innovation in SDVs security and privacy [20].

Supply chain complexity: the SDV supply chain comprises multiple layers of entities, including OEMs responsible for vehicle design and manufacturing, as well as suppliers across Tier 1, Tier 2, and Tier 3 levels. This layered structure increases complexity, as diverse software and hardware components from multiple suppliers must be integrated, often under heterogeneous cybersecurity standards [21, 80].

The analysis carried out in this work leads to the following contributions:

Comprehensive review: a complete review of the literature is presented, complemented by a questionnaire collecting insights from industry experts, to examine the current state of security and data privacy in SDVs while highlighting key challenges and emerging trends (Section 2.1).

Definition of SDVs: following [20], a definition of SDVs is provided to clearly distinguish among SDVs, AVs, and CVs (Section 3).

Attack surfaces and threats identification: potential attack surfaces and threats specific to SDVs are identified and categorized, taking into account their unique characteristics and technological innovations (Section 4.1).

Mitigation strategies: existing and emerging solutions for mitigating identified threats are examined (Section 4.2), with a particular focus on OTA updates (Section 5.1) and data privacy (Section 5.2).

Takeaway and future directions: open issues and future research directions are discussed, and a roadmap for enhancing the security and privacy of SDVs is outlined, with a dedicated summary box included in each section.

2 Methodology

The following two subsections present the methodology used to retrieve evidence and data from the systematic literature review and the industry expert elicitation.

2.1 Literature Review Process

The literature review for this study follows a systematic approach by adapting the SALSA (Search, Appraisal, Synthesis, and Analysis) method [23] to mine the relevant literature. The approach is based on the guidelines of Booth *et al.* [23] and is tailored to meet the specific context of SDVs. The process begins with the formulation of RQs and an exploration of background topics on SDVs. The output of this systematic process is a defined set of articles that address each RQ and topic. The workflow can be broken down into multiple stages of the method: Search, Appraisal, Synthesis, and Analysis. The search process is driven by the design of the RQs. This is followed by the development of the query to gather relevant literature from the selected primary and secondary sources.

Primary sources are databases that directly house academic research, while secondary sources provide aggregated or second-hand information and were used to supplement the search. The selected primary sources include relevant digital libraries such as IEEE Xplore [73], ACM Digital Library [3], Science Direct [129], and Springer Link [139]. The selected secondary sources include Google Scholar [58] and ResearchGate [121]. These secondary sources are known to return large volumes of results, and thus the search was limited to the first 100 results sorted by relevance. The query string "`software AND defined AND vehicle(s)`" was used across primary and secondary sources, with results filtered using Title-Abstract-Keywords (TAK) criteria for relevance. After retrieving the articles, the next phase involved assessing them according to inclusion and exclusion criteria. Inclusion required articles to be in English, peer-reviewed, and focused on security and privacy in SDVs. Exclusion criteria removed publications before 2004 and unrelated topics such as "Software-Defined Internet of Vehicles". A snowballing phase using the Paperfetcher Tool [107] followed, allowing backward and forward reference chasing to expand the selection. Due to the limited number of articles retrieved, additional tertiary sources were incorporated, including previous SDV surveys and articles recommended by the authors.

In the synthesis phase, selected papers were categorized by formulated RQs (Fig. 2). The figure illustrates the workflow of the literature review with the steps described earlier. The numbers in the circles indicate the number of articles retrieved after each filtering step and, ultimately, assigned to each RQ or topic. In the analysis phase, to answer the RQs, at least three authors independently reviewed the articles in each category and formulated their assessments. These evaluations were then consolidated, and the corresponding sections were written by the designated lead author.

2.2 Expert Elicitation

An expert elicitation methodology [29] complements the literature review and mitigates blind spots arising from the emerging nature of SDVs, where evidence remains sparse for some aspects (e.g., RQ1 and RQ4). The elicitation was designed as an exploratory expert study based on purposive sampling and limited to *industry participants*, aiming to capture high-signal practitioner judgments rather than population-level prevalence. To support replicability, the full questionnaire and anonymized

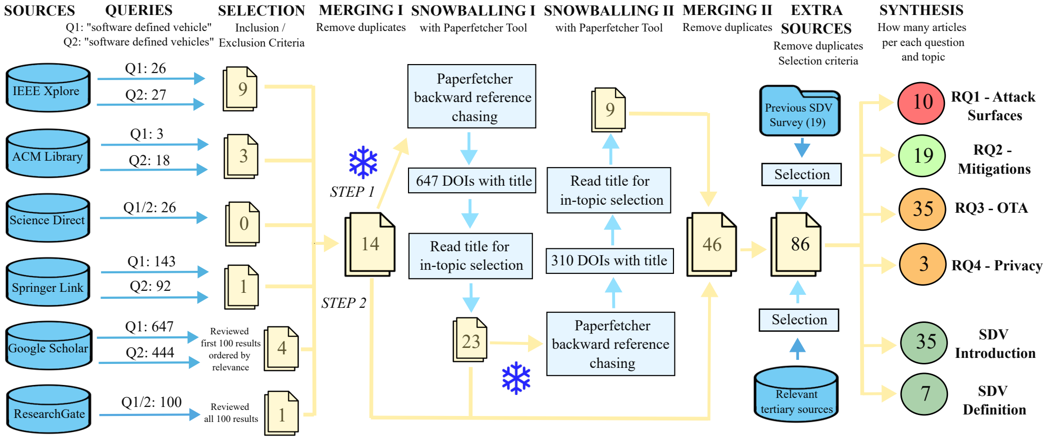


Fig. 2. Literature review workflow schema.

responses are publicly available[1], enabling reuse and validation. Table 1 summarizes the structure of the elicitation.

Elicitation design. The questionnaire was derived from the four research questions (RQ1–RQ4) and the foundational SDV definition goals. As summarized in Table 1, it includes 65 items organized into five sections: (i) SDV definition, (ii) attack surfaces and threats, (iii) mitigation strategies, (iv) OTA update risks, and (v) data privacy and governance. Most items used five-point Likert scales [89] with section-specific anchors. Prior to deployment, the questionnaire underwent internal content review to ensure coverage of the RQs and clarity.

Participant recruitment and composition. Experts were recruited via purposive sampling from the SDV supply chain, requiring professional involvement in automotive cybersecurity and exposure to at least one RQ. A targeted list of 22 experts ensured coverage across roles and perspectives; participation was voluntary and anonymous. Eleven complete responses were collected (50% response rate). Respondents represented OEMs (18.2%), Tier 1 (27.3%), Tier 2 (9.1%), and other automotive cybersecurity stakeholders (45.5%). Roles spanned engineering, technical leadership, senior expert, and executive positions.

Analysis methodology and metrics. Analysis employs descriptive, consensus-oriented measures suitable for exploratory Likert-scale studies. For each item, responses are summarized using the median and interquartile range (IQR), defined as IQR $= Q_3 - Q_1$. The median reflects typical expert judgment, while the IQR captures agreement. For example, the criticality of "Insecure Application Programming Interfaces (APIs)" (Q26) yielded a median of 5 and IQR of 1, indicating strong consensus. To complement these measures, Top-Two-Box Agreement (TTBA) is reported as the percentage of ratings equal to 4 or 5, providing an intuitive indicator of endorsement. Free-text responses were analyzed through lightweight thematic coding, revealing additional concerns such as fail-safe mechanisms and overlooked attacker profiles. As with all small-sample elicitations, results are indicative rather than statistically generalizable.

Presentation of elicitation results. Elicitation outcomes are reported in dedicated "Experts' feedback" subsections. Only representative items are reported and discussed, while all remaining items are made available online in the GitHub repository. Although limited in scope, combining expert input with existing research helps reveal both alignments and gaps between academic ideas and real-world perspectives, adding context to SDV security challenges.

---

[1] https://github.com/Marc-cn/sdv-security-privacy-elicitation

Table 1. Structure and characteristics of the expert elicitation questionnaire.

Section	Focus	# Items	Item Type	Scale / Format
Introduction	Participant background (organization type, role)	2	Categorical + free text	Multiple choice; open text
SDV Definition	Foundational SDV features and architectural principles	19 (+1 open)	Likert + open-ended	1–5 (not relevant → highly relevant)
Security Threats	Perceived criticality of SDV attack surfaces and risks	12	Likert	1–5 (low → high criticality)
Wireless Attack Vectors	Criticality of wireless interfaces	6	Likert (matrix)	1–5 (not relevant → highly relevant)
Possible Attackers	Relevance of attacker profiles	4 (+2 open)	Likert + open-ended	1–5 (not relevant → highly relevant)
Security Mitigations	Priority of technical, organizational, and regulatory mitigations	14 (+1 open)	Likert + open-ended	1–5 (low → high priority)
OTA Security	OTA risks, properties, and architectures	3 (+1 open)	Likert (grouped) + open-ended	1–5 (low → high criticality)
Privacy	Privacy governance, rights, and responsibilities in SDVs	7 (+1 open)	Likert + open-ended	1–5 (strongly disagree → strongly agree)
Overall	Expert elicitation across SDV lifecycle	**65 total**	Mixed (ordinal + qualitative)	Anonymous, self-administered

3 Definition and Foundational Concepts of SDV

The concept of SDVs is relatively recent, and existing definitions and key elements vary across authors and industrial stakeholders, with limited research focusing explicitly on SDVs. While several works introduce broad definitions followed by specific viewpoints, industry-oriented white papers and reports tend to emphasize SDV capabilities [134, 138]. As a result, a clear consensus on the most important SDV features has yet to emerge.

To establish a common baseline, we adopt the definition provided by some of the authors in [20]: *"SDV is an in-vehicle solution that enables abstraction and management of vehicle hardware components through software to create a scalable architecture with centralized and distributed local and remote controls. Furthermore, all vehicle software components must support OTA updates"*. SDVs embody a software-centric paradigm in which vehicle functionality is continuously introduced and evolved through software, requiring new architectures to meet scalability demands while significantly expanding the attack surface and elevating security and privacy to core design concerns.

Starting from this conceptual foundation, the main SDV features are identified through a combination of literature analysis, domain knowledge, and contributions from a dedicated expert-based questionnaire. The experts reviewed the initial list and provided feedback, which was integrated into the final version. The resulting features are organized hierarchically and grouped into three main categories: (a) decoupling of hardware and software, (b) smart vehicle and paradigm shift, and (c) transformation of automotive architectures and enabling technologies. Related notions are then examined, and similarities between SDVs and other devices or technologies are highlighted to further define and contextualize the SDV concept.

3.1 Core Features of Software-Defined Vehicles

This section details the core features that characterize SDVs, expanding on the three main categories introduced above. These features capture the architectural, functional, and systemic changes enabled by SDVs and form the basis for the subsequent analysis of security and privacy challenges.

(a) Decoupling of Hardware and Software. It represents a fundamental shift in vehicle design, where software increasingly defines functionality and hardware assumes an abstracted supporting role, enabling flexibility, scalability, and continuous evolution compared to traditional hardware-centric models [91, 125, 133]. In this paradigm, vehicles are primarily software-defined, with hardware providing execution and sensing, enabling modularity, upgradeability, dynamic resource reassignment, and runtime deployment or migration of services without hardware changes. The following list summarizes the main related features identified in the literature.

Hardware becomes an abstract shared resource: Software can be called/accessed, allowing a flexible combination that enhances both functionality and performance [91]. Furthermore, hardware abstraction plays a crucial role in effectively managing underlying hardware functions and application services such as described in the AUTOSAR Adaptive Platform with its Service-Oriented Architecture (SOA) [9]. This can be accomplished through the use of standardized interfaces to abstract hardware specifics, enabling developers to design software and applications without having to adapt to the unique characteristics of each hardware model. [90, 91].

Hardware and Software agnosticism: Ideally, software services and interfaces (such as APIs and middleware) should be hardware-agnostic to allow interaction of applications with other hardware services or communication with other applications [133]. This approach enables the sharing of solutions among manufacturers, suppliers, and academia. Software applications can be developed independently and tailored to specific functions or services and not to vehicle types [14].

Hardware/Software updateability: Vehicle functions and capabilities, powered by software, can be upgraded and managed continuously throughout the vehicle's entire lifecycle. These updates can also occur after sale [91, 133]. All vehicle components must support continuous OTA software updates, allowing for the agile [133] addition of new features, fixing vulnerabilities or software bugs, and optimizing existing functions [14, 91, 125]. This adaptability also enables vehicles to stay current and extend their useful life. As vehicle technology becomes increasingly software-centric, the reliance on software introduces challenges similar to those faced by computers and smartphones, such as software obsolescence, hardware compatibility issues, cybersecurity risks, and higher maintenance costs.

Enhancement of Software reuse: Reusing software components across different systems and platforms enhances development efficiency, reduces costs, and can ensure quality consistency [90].

(b) Smart Vehicle and Paradigm Shift. SDVs enable a paradigm shift in which vehicles become smart, data-centric systems capable of learning, adaptation, and continuous improvement. Through extensive sensing, connectivity, and computation, SDVs unlock data value across the vehicle lifecycle and integrate vehicles into a broader digital ecosystem. With increased integration of sensors, devices, and computing and communication capabilities, SDVs can manage and enhance data quality through closed-loop systems, enabling onboard software to support self-learning and evolution using AI. This development increases personalization of user experiences and supports large-scale data processing by connecting vehicles to cloud platforms [91]. Moreover, this shift positions the vehicle as part of a broader ecosystem, where customer-centric development continuously refines features based on real-world usage [14].

(c) Transformation of Automotive Architectures and Enabling Technologies. The transition toward SDVs is not limited to vehicle functionality, but is driving a broader transformation of the automotive industry. Traditional vehicle designs and development models are no longer sufficient to support SDV requirements. In this context, architectural and connectivity aspects emerge as key enablers of SDVs. In particular, the increasing complexity and volume of software require greater computing power [91] and increased security measures [14]. Hardware, software, and communication architectures must evolve accordingly.

Architecture: SDVs do not mandate a specific underlying architecture; however, zonal architectures align well with SDV objectives, driving a shift from domain-centric to zonal-centric E/E designs [18, 125]. Thus, systems are organized by physical location rather than function or domain. Each zone is managed by a dedicated controller, with zonal controllers interconnected and communicating with a central computing system. Additionally, the vehicle architecture must support scalable resources to accommodate the requirements of Vehicle-to-Everything (V2X) communications.

On-board operating system: This is a strategic component that enhances product competitiveness by providing integration and flexibility for vehicle control and application services. This system will evolve from partially integrated systems to comprehensive vehicle-level systems that support rich application ecosystems, improve control flexibility, and reduce integration costs [91].

Automotive Ethernet: It seems to be a good candidate for backbone communications in SDV architecture where a large amount of data is transmitted [39].

Cloud-native design paradigm: These paradigms should be adopted across hardware platforms from the cloud to the vehicle-edge, following an automotive DevOps (Development Operations) perspective [133].

Taken together, these core functionalities transform the vehicle into a continuously evolving software platform rather than a static product. While this shift enables unprecedented flexibility and innovation, it also requires rethinking security, safety, and governance across the entire lifecycle.

To conclude, the transition toward SDVs is often compared to the evolution of smartphones, reflecting a shift toward software-defined functionality and continuous feature updates [14, 91, 133]. While this analogy reflects the increasing role of software, abstraction, and OTA updates, automotive systems face challenges that differ from those of consumer electronics. In particular, SDVs must handle mixed-criticality workloads that combine safety and real-time constraints with high computational demands. They also resemble robotic systems in their reliance on frameworks, libraries, and APIs to manage complex sensor–actuator networks. A representative example is Apex.OS, a Robot Operating System (ROS) 2 variant developed by Apex.AI, which adapts robotics-oriented architectures to automotive systems [14]. Finally, SDVs adopt concepts from Software-Defined Networking (SDN): by decoupling network control from data forwarding, SDN principles enable centralized management, efficient communication, and dynamic resource allocation, supporting real-time coordination and cloud-based interaction [31].

3.2 Experts' Feedback

Focusing on the most relevant items from the elicitation (with full results available in the GitHub repository, as noted in Section 2.2), the highest level of expert consensus was observed for the decoupling of hardware and software, which received a median rating of 5 with low dispersion (IQR = 1) and a Top-Two-Box Agreement (TTBA) of 90.9%. These values indicate both strong endorsement and tight agreement among experts on its fundamental role in enabling flexibility, scalability, and modularity in SDVs. This result aligns with the emphasis found in the literature and reinforces its characterization as a core SDV feature. Other architectural enablers also exhibit high levels of endorsement. Automotive Ethernet received a median rating of 4, with low dispersion (IQR = 1) and a TTBA of approximately 72.7%, reflecting broad recognition of its importance as a communication backbone for SDVs. Similarly, continuous OTA updates and advanced data management were rated highly (median = 4–5, IQR = 1), with TTBA values above 80%, highlighting their perceived relevance for maintaining vehicle functionality and supporting long-term software evolution. In contrast, features related to generic software platforms show greater dispersion, with a higher interquartile range (IQR = 2) and a lower TTBA (approximately 63.6%). This pattern suggests

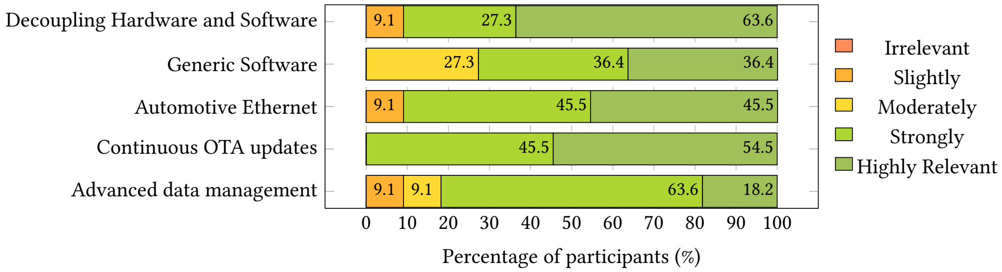


Fig. 3. Experts' answers for the definition section.

that while cross-platform and reusable software is widely acknowledged as relevant, experts differ more strongly in how they prioritize it relative to foundational architectural features.

Takeaway 1. The concept of SDVs is still evolving. This analysis reveals variations in the definitions of key elements across authors and industrial stakeholders, as well as a limited body of research explicitly dedicated to SDVs. As a result, a set of core SDV features is identified and their relevance is discussed based on both the literature and expert insights. This characterization provides a foundation for examining the associated security and privacy implications, while also offering guidance on aspects of SDVs that are likely to gain importance in the near future.

3.3 Differences between AV/CV/SDV

Although these three terms are frequently used interchangeably, AVs, CVs, and SDVs refer to distinct, yet partially overlapping, categories of vehicles.

AV refers to vehicles capable, at different levels, of driving without human intervention, regardless of whether their design includes controls for a driver [102]. AVs may operate across different levels of driving automation, as defined by the SAE J3016 standard, ranging from Level 0 (no automation) and driver assistance (Levels 1–2) to conditional, high, and full automation (Levels 3–5), with the ability to transition between autonomous and human-controlled operation depending on system capabilities and operational conditions [74, 161]. These vehicles use cameras and sensors to monitor the road and surroundings and make driving decisions, leveraging artificial intelligence to handle driving tasks, manage traffic scenarios, and prevent accidents. AVs may also communicate with other vehicles, using visual signals such as blinkers, and dedicated protocols to coordinate maneuvers safely [1].

CVs are vehicles equipped with V2X communication technology, enabling interaction with other vehicles, infrastructure, and road users through back-end systems that facilitate data exchange [150]. This includes Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, which use dedicated short-range radio signals to share information on vehicle speed, position, and direction, as well as road conditions and other relevant data [150]. The concept of connected vehicles dates back to the mid-1990s, and many contemporary vehicles already fall into this category. Connectivity can be considered an enabling factor for both AV and SDV functionalities.

SDV is the most recent of these three concepts and, as described in this section, refers to vehicles in which hardware components and functionalities are managed by software [20]. Beyond architectural implications already described, SDVs also enable new business models, such as subscription-based access to vehicle functionalities [72]. Depending on the features that the driver intends to use, specific vehicle functionalities can be enabled for a predetermined fee. This model allows users to pay only for the features they actually need at a given time. For instance, during extended periods of vehicle inactivity, users may temporarily disable advanced autonomous driving or comfort features, thereby avoiding subscription costs when such functionalities are not required. In SDVs, enabled and disabled features can range from an advanced navigation system to Level 3 autonomous driving and even performance-related settings such as speed or torque limiters tailored to different driving styles. Since these features are enabled or disabled through OTA configurations, SDVs typically require continuous network connectivity and therefore often also qualify as CVs.

SDVs differ from AVs and CVs by adopting a software-centric control and evolution model in which vehicle functions are centrally orchestrated and continuously upgradable, enabling faster innovation and new maintenance and business models compared to function-bound Electronic Control Units (ECUs). This paradigm naturally underpins scalable connectivity and higher levels of automation by supporting coordinated software control, data fusion, and cross-domain integration.

In summary, SDV, AV, and CV represent distinct yet overlapping characteristics of modern vehicles. Most vehicles today qualify as CVs, while those equipped with varying levels of Advanced Driver-Assistance Systems (ADAS) or autonomous driving features fall under the category of AVs. On the other hand, SDVs blur the lines by integrating software-managed features, potentially including autonomous driving as a configurable option. While a vehicle can be connected without being autonomous (CV only), or autonomous without being connected (AV only), such configurations are increasingly uncommon and we do not have clear evidence of this choice. However, SDV often require connectivity, though some configurations could be made through physical connections to the vehicle. This interplay highlights the nuanced relationships among these categories and reflects the ongoing evolution of vehicle technologies.

4 SDV Attack Surfaces and Mitigations

In this section, the security of SDV is discussed by addressing two interconnected research questions: **RQ1** focuses on identifying attack surfaces and threats (Section 4.1), while **RQ2** examines how the discovered threats can be mitigated (Section 4.2), with their interrelation discussed in Section 4.3.

4.1 RQ1 - What are the attack surfaces and associated threats in SDVs?

SDVs expose multiple attack surfaces that span software, communication, and data-management layers. Each identified attack surface is assigned a unique identifier (S1, S2, . . .) to allow concise referencing throughout the text and to link it explicitly to the corresponding threats (T1, T2, . . .) and mitigations discussed in Section 4.2. It is essential to note that the focus is limited to attack surfaces related to the specific characteristics of SDVs. Legacy attack surfaces inherited from autonomous and connected vehicles are nonetheless acknowledged, including S0-1 in-vehicle networks (e.g., Controller Area Network (CAN)) [11, 19], Automotive Ethernet [4, 39], S0-2 V2X communication [56], and S0-3 smart sensors [27]. These surfaces are still relevant to SDVs and must be considered. Fig. 4 summarizes the possible attack surfaces and includes several layers of interaction. Critical components such as zonal controllers, central computing units, and external third-party libraries are highlighted as key vulnerabilities, exposing the SDV to various cybersecurity risks from the supply chain to OTA updates. Based on the literature review, the following attack surfaces and related threats should be considered. They are interpreted in light of the defining characteristics of SDVs which contribute to multiple and sometimes overlapping points of exposure.

S1 - Insecure APIs: APIs in SDVs enable interaction between software components across in-vehicle and backend systems, but the exposure significantly enlarges the attack surface. Misconfigured APIs can allow authentication bypass, unintended access to production endpoints from development environments, and the invocation of backend services without valid access tokens [45]. Hard-coded environment configurations and client-side logic enabled attackers to switch from

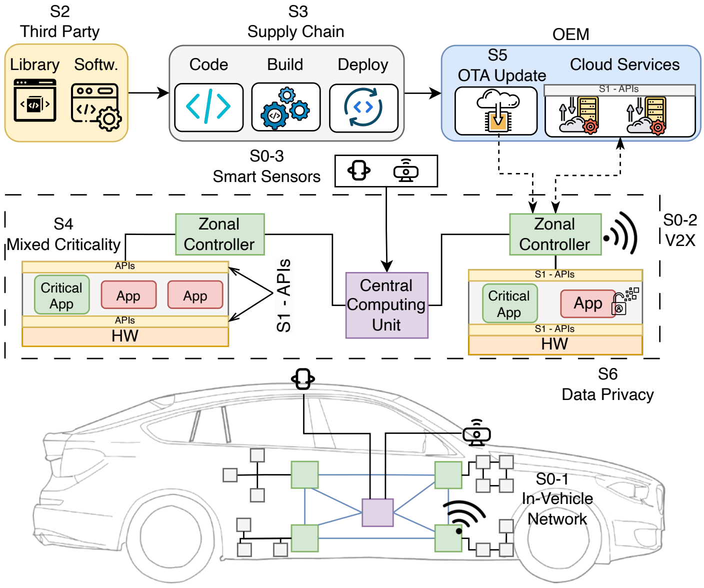


Fig. 4. Attack surfaces taxonomy, including both legacy (S0-1, S0-2, S0-3) and SDV attack surfaces (S1-S6).

development to production APIs, bypass corporate login mechanisms, and issue unauthenticated requests. This resulted in large-scale exposure of sensitive customer data, demonstrating how insecure API design can directly lead to data exfiltration and privacy violations. As a recent real-world example, the 2024 Kia connected-vehicle breach demonstrated how authentication bypass in exposed backend APIs enabled unauthorized access to vehicle functions and sensitive customer data [34]. *Main Threats:* client-side authentication bypass (**T1**) [44]; unauthorized production APIs access (**T2**); sensitive customer data exfiltration (**T7**) [45].

S2 - Third-party Applications and Libraries: SDVs increasingly depend on third-party applications, frameworks, and software libraries to accelerate development, promote reuse, and support rapid deployment. While this improves modularity and reduces time-to-market, it introduces a significant attack surface, as externally sourced components may contain vulnerabilities or malicious functionality. Evidence from adjacent domains, including mobile platforms [47, 116] and industrial control systems [36], shows that third-party software can be exploited to inject malicious code, bypass security mechanisms, or exfiltrate sensitive data. In the automotive context, inadequate vetting, missing integrity verification, or insecure update mechanisms for third-party components can enable unauthorized access to vehicle subsystems, exposure of personal and vehicular data (e.g., location and usage patterns), or interference with core vehicle functions, ultimately affecting safety and reliability [70]. *Main Threats:* dynamic malicious code injection (**T3**) [116]; distribution of malicious framework updates [116] (**T4**); sensitive customer data exfiltration (**T7**) [47].

S3 - Supply-Chain Security: Unlike S2, which focuses on runtime third-party software dependencies, this attack surface is broader and concerns the integrity of the entire production and distribution pipeline. SDV ecosystems depend on complex global supply chains spanning hardware manufacturing, firmware development, and software integration, introducing multiple points of trust violation. Prior work highlights how adversaries can exploit these dependencies to inject malicious artifacts into otherwise legitimate components, either by introducing counterfeit hardware [61] or by compromising build, packaging, signing, or distribution workflows [83]. In automotive systems, compromised supply-chain elements can undermine both safety and security guarantees, as malicious modifications may remain dormant until deployment or activation under specific conditions. *Main Threats:* distribution of tampered firmware updates or compromised software build (**T4**) [83]; counterfeit and recycled hardware components (**T10**) [61].

S4 - Mixed Criticality: SDVs increasingly adopt a SOA to support dynamic service deployment and flexible interaction among software components with heterogeneous criticality levels [93, 124]. While this paradigm improves modularity and updatability, it weakens the strict isolation traditionally enforced in safety-critical automotive architectures. Prior work shows that shared communication channels, middleware, and computational resources can cause unintended interference between services of different criticality [87]. In particular, unregulated service discovery, runtime binding, and resource contention may allow low-criticality services to affect the availability or integrity of high-criticality functions, potentially violating real-time and safety guarantees. These risks are concentrated at zonal controllers and central compute units, which host or coordinate multiple services and become high-value targets. *Main Threats:* unauthorized service interaction and access (**T2**) [124]; service-level Denial-of-Service (DoS) (**T5**) [124]; cross-criticality resource interference (**T6**) [87].

S5 - OTA Update: OTA updates enable remote software deployment across vehicle components, reducing maintenance costs and recall overhead. However, the OTA channel constitutes a safety-critical attack surface, as it requires privileged access to in-vehicle networks and ECUs [62]. Weak protection of update authenticity, integrity, or confidentiality can allow adversaries to inject or modify firmware during transmission or installation. Recent work demonstrates that compromised firmware can be customized and installed remotely, enabling persistent control over infotainment systems [33]. In addition, compromised update servers or distribution infrastructures may propagate malicious updates at scale, while insufficient version control exposes vehicles to downgrade and rollback attacks that reintroduce known vulnerabilities [8]. **RQ3** (Section 5.1) provides a detailed discussion of this topic. *Main Threats:* authentication bypass (**T1**) [34]; malicious firmware injection via OTA (**T3**) [62]; large-scale distribution of compromised updates (**T4**) [33]; rollback to vulnerable firmware versions (**T9**) [8].

S6 - Data Privacy: With the advent of SDVs, large volumes of data are continuously collected through onboard sensors and telematics units for diagnostics, service provisioning, and monetization by OEMs and third-party entities. As connectivity and data sharing increase, privacy risks intensify, since vehicular data often contains Personally Identifiable Information (PII) [55, 113], including precise geolocation and fine-grained behavioral information. Privacy attacks primarily target the misuse or extraction of such data by (i) unauthorized external parties or (ii) authorized entities exceeding their intended access scope or purpose [111]. These threats challenge consent, transparency, and regulatory compliance, particularly under data protection frameworks. **RQ4** (Section 5.2) provides a detailed discussion of this topic. *Main Threats:* unauthorized secondary use of personal data [111] (**T7**); location and trajectory inference attacks [55] and driver re-identification from vehicular data (**T8**) [55, 113].

4.1.1 ***Experts' Feedback*** As shown in Fig. 5, experts provided feedback on various security risks related to SDVs. The highest consensus was observed for Insecure APIs (S1), which exhibits a high median rating, low dispersion (IQR = 1), and a TTBA of 90.9%, highlighting significant concerns about APIs vulnerabilities. Similarly, malware in third-party applications and libraries (S2) was rated 4 or 5 by 45.5% of the experts, underlining the risk posed by untrusted software and emphasizing the need for stringent verification processes. Concerning supply chain security (S3), the experts' feedback was divided into hardware and software supply chain risks to capture insights in both areas. For hardware supply chains, 36.4% of experts rated risks as highly relevant, with

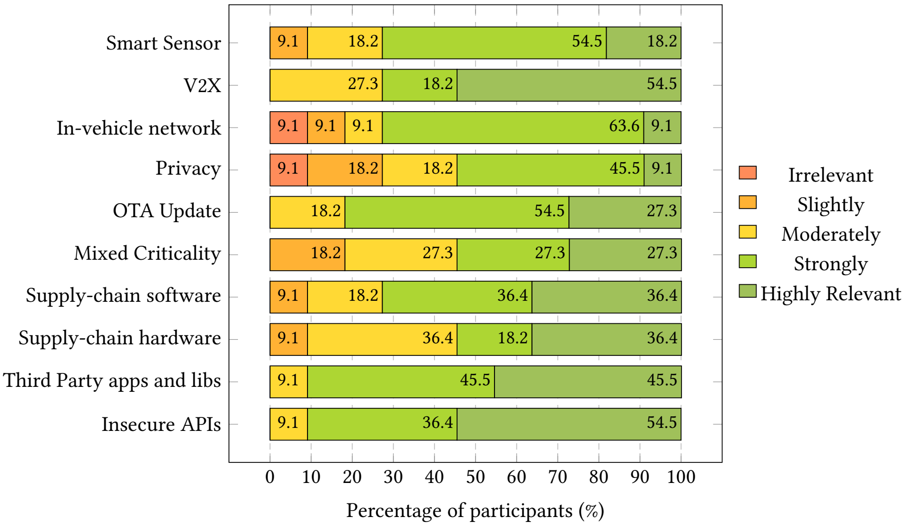


Fig. 5. The experts' answers to RQ1, where a rating of 1 indicates irrelevance and 5 indicates high relevance for SDV security.

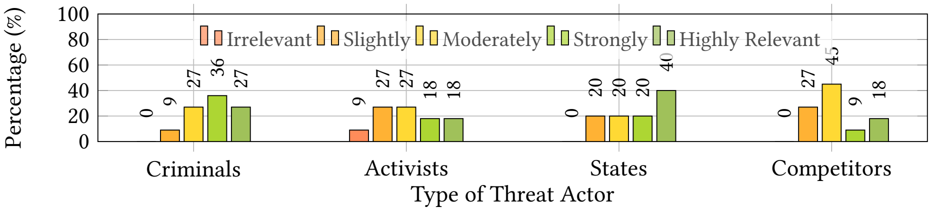


Fig. 6. Expert responses on the relevance of various attackers for SDV.

responses showing higher dispersion (IQR = 2). Regarding software risks, such as code injection, responses were more evenly distributed (IQR = 2), indicating that while both threat categories are acknowledged, perceptions of their criticality vary. Mixed-criticality attacks (S4) were perceived as moderately concerning, with a median rating of 3 and higher dispersion (IQR = 2), suggesting that this issue is considered relevant but less critical than other attack surfaces. In contrast, OTA updates (S5) were rated as highly critical by a majority of experts (TTBA > 70%, IQR = 1), highlighting the importance of securing update mechanisms. Privacy-related risks (S6) received moderate ratings with noticeable dispersion (IQR = 2), suggesting that privacy is often perceived as secondary to security concerns in SDVs. Finally, legacy automotive attack surfaces remain critical for SDVs. In-vehicle networks (S0-1) received high ratings with strong agreement (IQR = 1), while V2X communication (S0-2) was rated as highly critical (score 5) by 54.5% of experts. Smart sensors (S0-3) also received high ratings with low dispersion.

As shown in Fig. 6, expert feedback also highlights who the potential attackers of SDVs may be. Criminal actors and nation-states were perceived as the most significant threats, both receiving high ratings with low dispersion, while activists are perceived as the least relevant group. Open-ended responses further identified additional attacker profiles, including insiders, security researchers, and low-sophistication adversaries (e.g., script kiddies), suggesting a broader and more heterogeneous threat landscape.

Takeaway 2. Existing attack surfaces in CVs and AVs, such as V2X communication and in-vehicle networks, will continue to be critical in SDVs. Emerging attack surfaces such as insecure APIs, third-party software libraries, and supply chain vulnerabilities will introduce additional challenges unique to SDVs. Addressing these emerging attack surfaces alongside traditional ones is key to safeguarding SDVs as they evolve.

4.2 RQ2 - What strategies can mitigate attack surface vulnerabilities?

As discussed in Section 4.1, the attack surfaces of SDVs span from unauthorized data access to the takeover of critical vehicle functions. This section outlines strategies to mitigate vulnerabilities across these surfaces. Rather than isolated countermeasures, the proposed mitigations are framed as design guidelines for securing SDVs, differing in architectural assumptions, lifecycle adaptability, and trade-offs such as performance overhead and certification complexity. The proposed mitigations are grouped into overarching categories. These classifications are based on solutions found in the literature and feedback from industry experts gathered through the questionnaire. Similarly to the attack surface categorization in the previous section, each mitigation category is assigned a unique identifier (M1, M2, ...) to facilitate easy reference in the text and to the corresponding attack surfaces and threats.

M1 - Intrusion Detection and Prevention Systems (IDPSs): SDVs can employ IDPSs to mitigate attack surfaces such as malware injection via third-party applications (S2) by monitoring internal and external communications for anomalies [84, 131, 151]. These systems detect unauthorized access and abnormal data flows, enabling early response. However, practical deployments suffer from high false positive rates, performance overhead, and attacker evasion. Improving accuracy and robustness therefore remains an open research challenge [95, 114]. AI-based Intrusion Detection Systems (IDSs) for In-Vehicle Networks (IVNs) apply Machine Learning (ML) and Deep Learning (DL) to analyze communication patterns, including the CAN bus, and detect both known and unknown attacks [120]. Nevertheless, simpler models often provide stronger baselines, as learning-based approaches trained on synthetic data may overfit and fail to generalize to real-world conditions [7]. Expert responses highlight the importance of Gateway Firewalls [112], rated as highly critical by 63.6% of participants, which, when combined with IDPSs, enable traffic filtering and real-time intrusion response. From a design perspective, deploying centralized IDSs at gateways or high-performance computing (High-Performance Computing (HPC)) nodes provides global system visibility but introduces potential single points of failure, while distributed IDSs at the ECUs improve fault containment at the cost of increased management complexity. AI-based IDSs further enhance adaptability, but raise challenges in explainability and certification, especially when trained on synthetic data [153].

M2 - Secure Software Development Practices: A key mitigation strategy is the Software Vehicle Security Engine (SVSE) Lifecycle for API and supply-chain security (S0-1, S3). Applying secure coding practices throughout SDV software development reduces vulnerabilities introduced at design and implementation time [101]. This includes systematic testing through static and dynamic analysis, fuzzing, and penetration testing. Adopting a secure Software Development Life Cycle (SDLC) [101, 106], with security embedded at all stages, is essential. Such a lifecycle incorporates threat modeling and risk assessment [65, 85, 99] to identify threats (e.g., spoofing, tampering, information disclosure), supported by automated frameworks such as Threat Analysis and Risk Assessment (TARA) [101]. Secure and standardized vehicle APIs are also required [64, 157]. To mitigate third-party software risks (S2), Software Composition Analysis (SCA) can identify vulnerabilities in external libraries and open-source components before deployment [106, 160]. Expert feedback confirms the relevance of the SVSE life cycle, with 54.5% of respondents rating it as highly critical. In SDVs, secure software development extends beyond pre-deployment and becomes a continuous requirement, as OTA updates, third-party integrations, and cloud services expand the attack surface over the vehicle lifecycle. Complementing SCA, the use of Software Bills of Materials (SBOMs), such as in UPTANE [142], can improve supply-chain security by providing component-level transparency and supporting vulnerability disclosure and patch management for third-party software [100, 146].

M3 - Automotive Ethernet Security: In our survey, Automotive Ethernet has been identified as a key enabler for SDVs. However, Automotive Ethernet (AE) can inherit the vulnerabilities of standard Ethernet [39]. To mitigate security risks in AE, several strategies have been proposed to address its vulnerabilities. Firewalls can control data flow between networks, blocking unauthorized access [117]. IDSs monitor network activities in real-time, identifying and preventing potential threats. Network and link-layer security mechanisms such as IPsec and MACsec are essential for ensuring data confidentiality and integrity, particularly when transmitting sensitive information within the vehicle [37, 39]. Furthermore, Transport Layer Security (TLS) protects point-to-point communications, mainly securing unicast transmissions. VLANs play a key role in segmenting the network, reducing the risk of lateral movement by isolating sensitive domains [78, 136]. Timed Efficient Stream Loss-Tolerant Authentication (TESLA) and Secure ARP (S-ARP) add further protection by preventing replay attacks and Address Resolution Protocol (ARP) poisoning [162]. Together, these mitigation techniques can increase Automotive Ethernet security with a relative impact on network performances.

M4 - OTA Defenses: Both the literature and questionnaire responses emphasize a multi-layered approach to OTA security, spanning in-vehicle and backend components (Section 5.1). Effective defenses include secure boot, authenticated and integrity-protected update packages, digital signatures, role-based access control, and continuous logging and auditing. Together, these mechanisms ensure that updates are delivered, verified, and installed in a controlled and traceable manner. Encryption and authentication must be enforced throughout the update lifecycle to prevent interception, replay, or manipulation. In particular, update packages should be digitally signed and verified by the vehicle prior to installation, ensuring that only software from authorized sources is accepted and reducing the risk of malicious code injection. Lightweight public-key schemes based on elliptic curves, such as those using Elliptic Curve Cryptography (ECC), are well suited for resource-constrained vehicular environments and are commonly used for key exchange and signature generation [20, 122]. More decentralized approaches move verification and rollback mechanisms into the vehicle, improving resilience while complicating key management and certification.

M5 - Data Anonymization: Anonymization techniques are critical for protecting data privacy in SDVs while preserving data utility. Data masking transforms sensitive attributes, such as precise GPS coordinates or driver identifiers, through pseudonymization or obfuscation. K-anonymity [141] ensures that each data record is indistinguishable from at least $k-1$ others, reducing re-identification risks. In SDVs, this can be applied to driving patterns or route data by aggregating records into similarity groups, providing cohort-level indistinguishability. L-diversity and t-closeness extend k-anonymity to address attribute disclosure. L-diversity enforces diversity of sensitive values within each group, such as driver behavior or location patterns, while t-closeness further strengthens privacy by ensuring that the distribution of sensitive attributes (e.g., fuel consumption or braking patterns) closely matches that of the overall dataset [88].

M6 - Privacy-preserving Techniques: Privacy-preserving techniques, such as Differential Privacy (DP), can provide a mathematical framework for protecting individual privacy in SDVs while preserving data utility for analytics on aggregated vehicle datasets [104]. SDVs generate high-resolution data streams from in-vehicle sensors. For instance, by implementing differential privacy, controlled noise is systematically injected into aggregate query results, ensuring that the presence

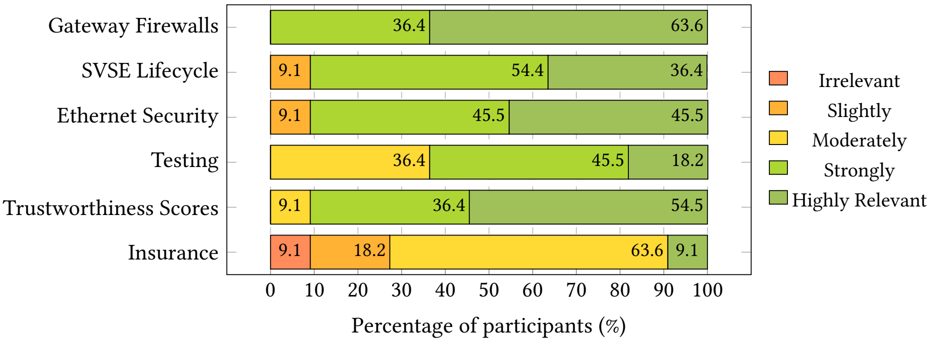


Fig. 7. The experts' answers to RQ2, where a rating of 1 indicates irrelevance and 5 indicates high relevance for SDV security.

or absence of any single data point has a minimal, statistically bounded impact on the output. This probabilistic noise addition obfuscates individual-level data, preserving privacy while enabling the extraction of meaningful patterns such as driving behaviors. Differential privacy enables these analyses to remain compliant with regulatory standards by providing quantifiable privacy guarantees while upholding data utility for the data processor.

M7 - Security-Aware Task Orchestration and Resource Allocation: In SDVs, this enables managing security workloads and reallocating tasks across ECUs to address cyberattacks and safety issues. Given limited ECUs resources, security mechanisms must be efficient. Hypervisor-based virtualization allows applications with different safety-criticality levels to share hardware while ensuring isolation, improving security and efficiency [108]. In addition, task migration and runtime orchestration can mitigate cyberattacks or safety faults by reallocating tasks across ECUs. These approaches include static methods [12, 156], where precomputed mappings for expected fault or attack scenarios are stored and activated at runtime, and dynamic methods [63], where migration decisions are made on the fly based on available resources and the affected tasks. Such mechanisms are especially effective in mixed-criticality systems, where attacks or faults may originate from components with different criticality levels. They help limit the impact of compromised tasks without requiring dedicated redundant hardware or static backups for every task.

4.2.1 ***Experts' Feedback*** Beyond the previously mentioned mitigations, industry experts suggest additional measures for improvement. We report some of the results of the questionnaire (Fig. 7).

M8 - Testing is a key area for improvement in SDVs: the adequacy of current testing methods received a median rating of 4 with low dispersion (IQR = 1) and TTBA = 63.6%, indicating broad (though not unanimous) endorsement. Automated testing frameworks, such as those that generate test cases from threat models (e.g., TARA [95]), are critical to ensure that cybersecurity measures are comprehensive and scalable. Experts emphasized the need for more efficient testing processes that can keep up with the rapid development and deployment cycles in SDVs. This includes integrating cybersecurity testing into continuous deployment pipelines, ensuring that security is continuously monitored throughout the software lifecycle [95, 101].

M9 - Trustworthiness Scores serve as a metric to evaluate overall security of automotive software. This approach involves calculating a composite score based on various security factors, such as code quality, patching frequency, and vulnerability severity. Trustworthiness scores can also help manufacturers demonstrate compliance with standards such as ISO/SAE 21434 [126].

M10 - Insurance is identified as a complementary mitigation strategy, with the literature reporting mixed views on its overall effectiveness [22, 135, 137]: experts mostly converged on moderate relevance (median = 3, IQR = 1), while strong endorsement was limited (TTBA = 9.1%). Although some respondents view insurance as a crucial countermeasure to reduce the financial impact of security breaches, others argue that it should not be seen as a substitute for strong security practices. Instead, insurance could complement robust security measures by providing a safety net for potential damages, while manufacturers focus on implementing proactive strategies to prevent breaches.

Takeaway 3: *Security-by-design principles* should guide the development and integration of each component, with continuous monitoring to detect evolving threats. This approach requires planning by OEMs to ensure secure vehicles in the coming years.
Takeaway 4: *Importance of a Multi-Layered Security Approach.* The interconnected nature of SDVs demands a multilayered security architecture, incorporating in-vehicle, edge, and cloud-based measures.
Takeaway 5: *Necessity of Standardized Protocols and Compliance with Regulations.* The current dependence of the industry on various protocols underscores the need for standardized approaches (e.g., following ISO/SAE 21434 and UNECE WP.29 regulations).
Takeaway 6: *Need for Robust Supply-Chain Security.* The SDV ecosystem's reliance on components and software from various suppliers introduces significant risks. A secure supply chain requires rigorous supplier screening, secure coding practices, and regular audits.
Takeaway 7: *Adoption of Trustworthiness Metrics and Continuous Improvement.* As SDVs continue to evolve, trustworthiness metrics, calculated based on code quality, patch history, and other factors, offer a valuable measure of system health.

4.3 Discussion

Table 2 and Fig. 8 formalize a security framework grounded in the literature review and expert elicitation by organizing heterogeneous vulnerabilities into a traceable chain from exposed interfaces (S1–S6), to adversarial objectives (T1–T10), and finally to enforceable controls (M1–M10). The visual schema captures these dependencies by aligning attack surfaces, threats, and mitigations in distinct columns and by annotating each mitigation with its corresponding *NIST Cybersecurity Framework (CSF) 2.0* action class (Identify, Protect, Detect, Respond, Recover, Govern) [103]. This mapping contextualizes SDV-specific controls within a widely adopted, regulation-agnostic cybersecurity taxonomy. NIST CSF 2.0, released in 2024, is particularly relevant in this context because it extends the original framework with an explicit *Govern* function and strengthens lifecycle-oriented risk management. This makes it complementary to automotive standards such as ISO/SAE 21434 and UNECE R155/R156, which emphasize continuous risk assessment and organizational accountability. Mapping the identified mitigations onto CSF 2.0 highlights that the most security-relevant trust boundary in SDVs remains the *software supply and update continuum*. The same threat classes that manifest at runtime through insecure APIs (S1) and third-party components (S2), including authentication bypass and unauthorized access (T1–T2), malicious code injection and compromised distribution (T3–T4), and sensitive data leakage (T7), also reappear within the supply chain (S3) and the OTA pipeline (S5).

As a consequence, effective mitigation cannot be addressed through isolated point solutions. Secure software development and SCA practices (M2) primarily support the Identify and Protect functions, reducing the introduction of exploitable artifacts. Automotive Ethernet security mechanisms (M3) further strengthen the Protect function by enforcing isolation, secure communication, and traffic control across in-vehicle networks. In contrast, OTA defenses (M4) enforce provenance, integrity, and anti-rollback guarantees (T9), effectively turning update distribution into a policy-controlled security perimeter spanning vehicle and backend services. Detection and

Table 2. Integrated attack Surface–Threat–Mitigation mapping.

Attack Surfaces and related Threats (Section 4.1)				Surface Mitigations (Section 4.2)	
SID	**Surface**	**TID**	**Threat Name**	**MID**	**Mitigation**
S1	API Security	T1	Authentication bypass	M1	Intrusion Detection and Prevention Systems
		T2	Unauthorized access	M3	Automotive Ethernet Security
		T7	Sensitive data exfiltration and privacy leakage	M4	OTA Defenses
				M6	Privacy-preserving techniques
S2	Third-party Apps and Lib.	T3	Malicious code or firmware injection	M1	Intrusion Detection and Prevention Systems
		T4	Distribution of compromised software	M2	Secure Software Development Practices
		T7	Sensitive data exfiltration and privacy leakage	M6	Privacy-preserving techniques
S3	Supply-Chain Security	T4	Distribution of compromised software	M2	Secure Software Development Practices
		T10	Hardware trust violation and counterfeit	M8	Testing
				M9	Trustworthiness scores
				M10	Insurances
S4	Mixed Criticality	T2	Unauthorized access	M1	Intrusion Detection and Prevention Systems
		T5	Service-level DoS and resource exhaustion	M3	Automotive Ethernet Security
		T6	Cross-criticality interference affecting safety	M4	OTA Defenses
				M7	Security-aware task orchestration
				M8	Testing
S5	OTA Update	T1	Authentication and authorization bypass	M4	OTA Defenses
		T3	Malicious code or firmware injection		
		T4	Distribution of compromised software		
		T9	Rollback or downgrade to vulnerable system states		
S6	Data Privacy	T7	Sensitive data exfiltration and privacy leakage	M3	Automotive Etherenet Security
		T8	User re-identification and inference	M5	Data Anonymization
				M6	Privacy-preserving techniques

prevention systems (M1) act as a cross-cutting Detect control that compensates for residual design and implementation flaws across S1 and S4 by monitoring service interactions and anomalous data flows. However, the framework makes explicit that detection alone cannot replace preventive guarantees where safety-critical behavior is implicated. Mixed-criticality exposure (S4) further highlights the need for evidence-driven separation. Testing and verification (M8), classified under Identify, must validate isolation properties, timing constraints, and resource budgets against cross-criticality interference (T6). Privacy-oriented mitigations (M5–M6) primarily populate the Protect function by constraining information release and bounding inference risk for T7–T8. Finally, governance mechanisms such as trustworthiness scores and insurance (M9–M10) align with the Govern function by incentivizing measurable security posture and risk transfer.

The NIST CSF 2.0 mapping also exposes a structural gap: across the surveyed literature, mitigations largely concentrate on Identify, Protect, and Govern, while explicit mechanisms addressing *Respond* and *Recover* remain scarce, despite being explicitly defined in the NIST CSF 2.0. This imbalance indicates that current SDV security research prioritizes prevention and assurance over resilience and post-incident handling. Only limited efforts, such as M4 through OTA-oriented defenses and, more complete, M7 via security-aware task orchestration [63], attempt to partially bridge this gap, which nonetheless remains largely understudied in software-driven and continuously evolving vehicular systems.

5 SDV Security Challenges

In this section, several critical security challenges arising from the previous RQs are examined. The focus is placed on OTA systems and methodologies, as well as the complexities of managing data privacy in SDVs. These challenges are derived from an in-depth analysis of current technological trends, newly identified attack surfaces, and vulnerabilities highlighted in the literature review.

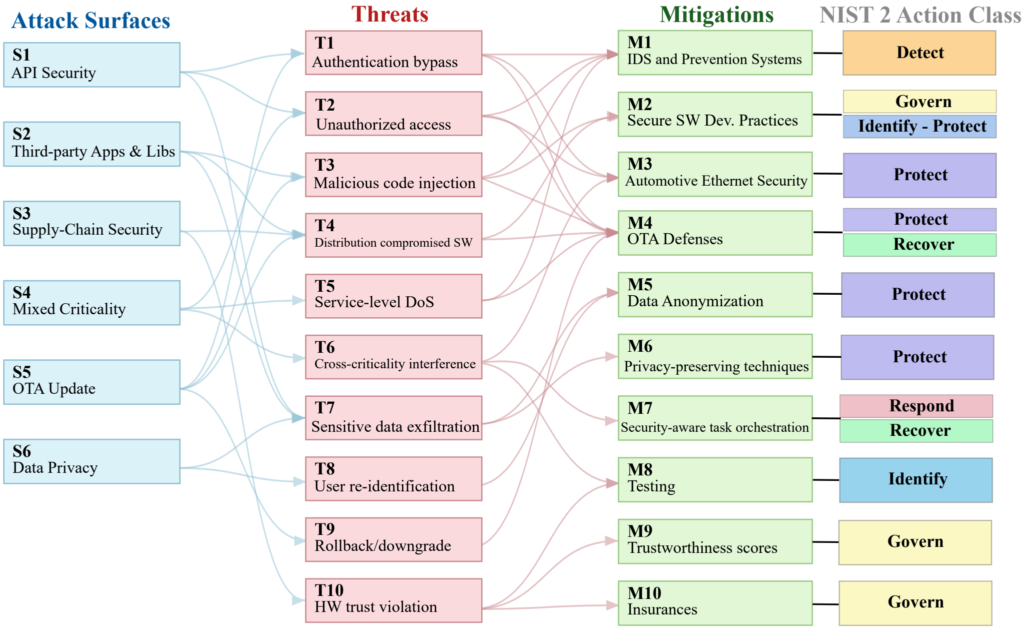


Fig. 8. Conceptual mapping between SDV attack surfaces, threat classes, and mitigation strategies.

5.1 RQ3 - What are the main security issues that OTA updates face, including the out-of-the-vehicle environment?

Updating vehicle firmware via OTA has become a crucial feature, as the software-driven nature of modern vehicles require continuous updates to maintain safety and security. In SDVs, OTA updates are a fundamental capability, enabling continuous feature delivery, bug fixes, and security patches. A major challenge in SDVs is implementing secure and efficient OTA mechanisms that address threats originating both inside and outside the vehicle, including malicious firmware introduced through supply chains. From a system-level perspective, OTA mechanisms in SDVs must be analyzed not only as communication protocols but as distributed, safety-critical update workflows spanning cloud backends, edge infrastructure, and in-vehicle components. These workflows involve multiple stakeholders, including OEMs, suppliers, and cloud service providers, introducing additional trust dependencies and coordination challenges across the SDV ecosystem. This introduces new constraints compared to traditional vehicular paradigms, including (i) coordination across heterogeneous ECUs with different criticality levels, (ii) real-time constraints that limit update latency and downtime, and (iii) the need to ensure consistency and atomicity of updates across interdependent software components. These aspects highlight that OTA security in SDVs is not only a cryptographic problem, but also a system orchestration and lifecycle management challenge. This creates a trade-off between update flexibility and system assurance, as increased update frequency and distribution complexity can amplify both operational risks and the attack surface. Although existing solutions aim to preserve the authenticity, integrity, and confidentiality of software components, they often rely on centralized, cloud-based approaches managed by OEMs. Such solutions may inadequately address SDV-specific challenges, including decentralized, real-time updates across multiple components with minimal operational disruption. To support software maintenance, OTA mechanisms have therefore evolved over the last decade. However, many approaches still lack essential security and privacy protections, such as hardware-backed

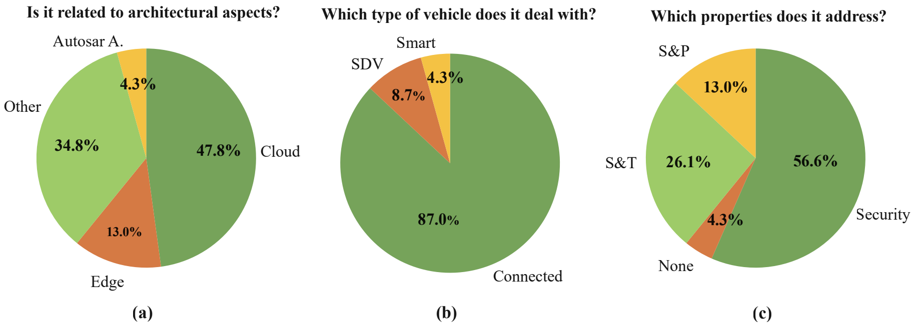


Fig. 9. OTA literature review outcomes: (a) Type of architecture, (b) Type of vehicles, and (c) Addressed properties.

secure boot chains to ensure firmware integrity or rollback protection against downgrade attacks. Since the 2000s, numerous supply-chain firmware attacks have been reported, e.g., a 2024 vulnerability in a Hyundai firmware update mechanism enabled integrity violations [33]. Such attacks remain particularly concerning, as malicious firmware can compromise hardware integrity before deployment, making detection difficult. The following section examines OTA mechanisms in the literature, analyzing their architectures, target vehicles, and the properties they aim to guarantee.

5.1.1 Literature Review.

A total of 35 articles, including research papers, white papers, surveys, and standards, were analyzed. After an initial screening, 23 papers were selected from the original set. The selection has been made by considering only papers that contribute to the advancement of the state of the art in OTA mechanisms. The literature review revealed that most existing OTA solutions focus on network security, addressing potential attackers who may try to intercept or disrupt network communications. In particular, 87% of the articles analyzed refer to *connected vehicles* more than to SDVs (8.7%) denoting that network security has been considered the most crucial challenge to address (Fig. 9).

According to the literature review (Fig. 9), 47.8% of proposed solutions adopt a cloud-based architecture for managing OTA processes, where software is delivered directly from OEMs to vehicles. Among these, [159] leverages blockchain to ensure authenticity and [140] proposes a blockchain-based architecture for secure OTA updates. In 2016, UPTANE was introduced as a dedicated protocol for OTA software updates [142]. Built as an extension of The Update Framework (TUF), UPTANE leverages strong cryptographic primitives to ensure the integrity, authenticity, and freshness of software updates, even under partial infrastructure compromise. It introduces a multi-role signing architecture using signed metadata, hash verification, and version constraints to mitigate threats such as machine-in-the-middle, replay, rollback, and malicious update injection. A key contribution is the separation between offline-protected signing authorities and online distribution servers. UPTANE also defines a two-tier trust model with a Primary ECU orchestrating updates and multiple Secondary ECUs locally verifying software images before installation. This design enables defense in depth, prevents unauthorized lateral update propagation, and supports fine-grained ECU-level authorization. Since 2016, UPTANE adoption has increased, leading to extensions such as in-toto [146] and Scudo [100], which focus on supply-chain verification and ECU update security, respectively.

Only 13% of OTA solutions for SDVs explore edge-based approaches, where updates are distributed via architectural edge nodes rather than a centralized entity. Works such as [94] and [16] address challenges including network availability, bandwidth, and software fragmentation.

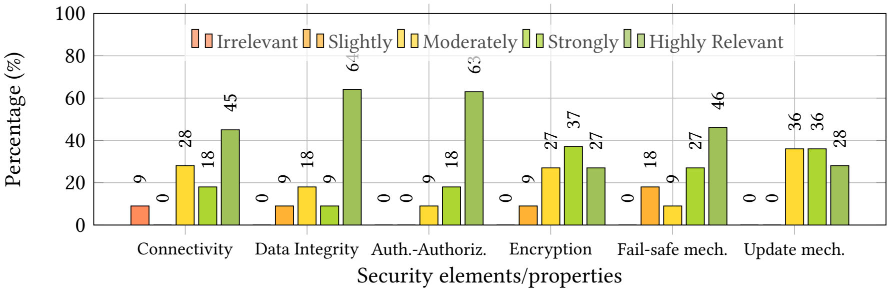


Fig. 10. Expert responses on the significance of security elements/properties in OTA updates.

However, [94] focuses solely on securing communication between the OEM server and the vehicle without leveraging SDV architecture, while [16] does not address security. A smaller share (4.3%) targets the AUTOSAR Adaptive Platform, which adopts a microservice architecture to support efficient software distribution. For example, [66] proposes a secure OTA firmware update mechanism using Message Queuing Telemetry Transport (MQTT), whose lightweight publish/subscribe model improves system dependability and client independence. Regarding the third subquestion (Fig. 9), most mechanisms address security (56.5%), while trust (26.1%) and privacy (13%) are considered mainly in combination with security rather than as standalone properties.

5.1.2 ***Experts' Feedback*** Consistent with the literature, security emerged as the most critical non-functional property of the OTA update process (IQR = 1, TTBA > 80%). Privacy was also identified as important, though with greater dispersion, indicating less uniform prioritization. In contrast to the literature review, experts did not favor centralized cloud-based solutions; instead, Public Key Infrastructure (PKI)-based mechanisms were rated as the most effective architectural option (IQR = 1, TTBA > 70%), while centralized approaches showed lower agreement. Experts further assessed the overall criticality of OTA updates in SDVs, rating risks such as disruption, rollback, and malicious firmware installation as highly critical (median = 5, TTBA > 80%). As shown in Fig. 10, strong consensus was observed on key OTA security elements, including Authentication and Authorization, Data Integrity, and Fail-Safe mechanisms (IQR = 1). Data Integrity received the highest criticality, followed by Authentication and Authorization, reinforcing the central role of security mechanisms in robust OTA update processes. These results confirm findings from the literature that identify security as the primary non-functional requirement for OTA updates.

Takeaway 8. As SDVs increasingly rely on OTA updates for software maintenance, securing these updates is essential. Data integrity, authenticity, and secure communication channels prevent unauthorized access and ensure only verified updates are installed. Encryption, digital signatures, and access controls mitigate rollback attacks and protect software during transmission. However, strong security mechanisms can affect the performance and sustainability of OTA processes. Effective OTA systems must balance low overhead with robust security, with future research exploring edge computing, advanced cryptography, and distributed architectures. Importantly, OTA security in SDVs must be addressed as a system-level problem involving architectural design, trust distribution, and lifecycle coordination, rather than as a set of isolated security mechanisms.

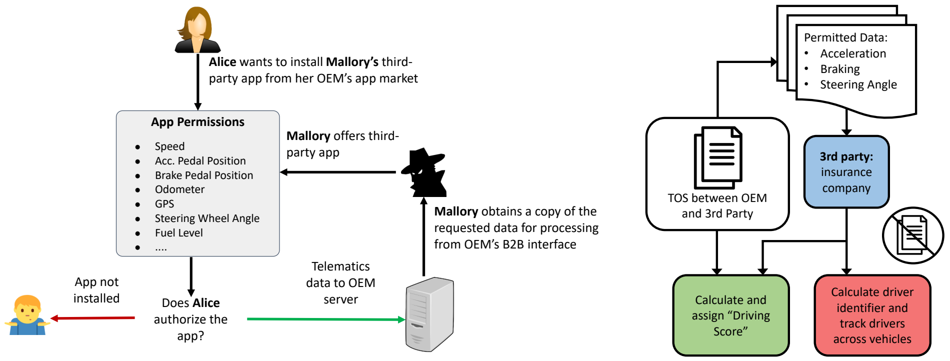
5.2 RQ4 - How do SDVs affect data collection, and what are the primary concerns related to user and vehicle privacy?

This section reviews prior work on privacy in automotive data collection, excluding studies that jointly address security and privacy since they primarily focus on security, as commonly observed in the broader V2X privacy literature. Vehicle data collection has been practiced for over a decade by both OEMs [57, 60] and third-party actors such as Usage-Based Insurance (UBI) providers [5, 6, 51, 118]. In both cases, only a selection of vehicle data was collected for vendor-specific purposes. These early forms of telematics data collection platforms were either (i) only available on upper-tier models of certain OEMs or (ii) as a hardware dongle that accesses publicly available sensor data through OBD-II. From a system perspective, data privacy in SDVs differs from traditional vehicular contexts due to continuous data generation, cloud integration, and cross-domain data sharing. Privacy risks therefore emerge not only from data collection, but also from data aggregation, inference, and secondary use across distributed ecosystems, third parties, and infrastructure providers. A distinguishing privacy challenge in SDVs arises from their ability to dynamically evolve data access capabilities through OTA updates. Unlike traditional connected vehicles, where data access pathways are largely fixed at design time (e.g., via static interfaces), SDVs enable OEMs or third-party components to introduce new software modules that can subscribe to previously inaccessible sensor data. For example, an OTA update may introduce new analytics services that subscribe to raw sensor streams (e.g., camera, location, or driver behavior data) that were not previously accessible to that component, enabling new forms of profiling or inference. This creates a form of evolving data access surface, where the set of collected data is not fixed but can expand over time, often without direct user awareness. Consequently, privacy risks are not limited to initial data collection, but extend to the continuous redefinition of what data is collected, how it is used, and by whom.

Automotive data privacy has gained increased scrutiny as vehicles have become valuable sources of personal information. Mozilla reported serious privacy violations across multiple automakers [53], finding that manufacturers such as Nissan, Volkswagen, and Toyota collect sensitive data including routes, ethnicity, weight, facial expressions, and sexual behavior. Analyses of Honda's opt-out mechanisms identified the use of dark patterns [25, 59, 98]. Further investigations revealed that General Motors shared driving behavior data with insurance companies via OnStar without adequate customer awareness, increasing insurance costs [67]. The emergence of platforms such as Android Automotive OS (AAOS), now adopted by several major OEMs (including Honda [68], General Motors [42], Ford [52], Volvo/Polestar [152], and Stellantis [81]), further amplifies these concerns. While AAOS inherits Android's permission model, its access to a broader set of in-vehicle sensors and its large-scale deployment increase the potential for privacy violations.

The surveyed literature largely addresses privacy through generic techniques and does not explicitly address two challenges that are specific to the SDV paradigm. First, *telemetry retention* is typically implicit: SDVs enable persistent, high-resolution data collection over the vehicle lifecycle (e.g., OTA logs and continuous diagnostics), raising long-term retention and secondary-use risks beyond those considered for traditional CVs. Second, *consent management* is usually assumed to be static, despite the dynamic and contextual nature of consent in SDVs, where features can be enabled or disabled via OTA and vehicles may be shared among multiple users. The limited treatment of these aspects highlights an open gap in current SDV privacy research. The following sections outline applicable privacy regulations for SDVs, review automotive privacy attacks, and discuss defense frameworks.

5.2.1 Regulations Early automotive privacy regulation relied on voluntary guidelines from the Alliance of Automobile Manufacturers (AAM) and the 2015 Driver Privacy Act within the FAST



(a) Data Collection in BMW CarData [60] (b) Data Sanitization [111]

Fig. 11. Data collection and sanitization explained.

Act in the US [113]. In May 2018, the European Union introduced the General Data Protection Regulation (GDPR) [49] as the first comprehensive privacy framework, strengthening individual consent and data protection. Although formally applicable to EU entities, GDPR compliance is essential for global OEMs operating across regions. In the United States, state-level laws such as the California Consumer Privacy Act (CCPA) [105] and the stricter California Privacy Rights Act (CPRA) [30] further shape automotive data governance. The GDPR distinguishes among *data controllers*, *data processors*, and *data subjects*, granting strong protections to drivers as data subjects. Data controllers bear primary responsibility for determining how and why data are processed, while processors act on their behalf. OEMs are classified as data controllers due to their role in managing data shared with third-party app providers, leading to heightened compliance obligations [158]. An evaluation of OEMs privacy policies revealed inconsistent and incomplete GDPR implementations [111]. As a result, three privacy goals were identified as essential for achieving GDPR compliance in automotive systems [111]:

Data Minimization: Current telematics platforms such as BMW CarData [60] and AAOS rely on permission models to limit access to specific vehicle sensors. The BMW CarData data flow, illustrated in Fig. 11a, is similar to that of AAOS. However, the AAOS permission model lacks sufficient granularity, grouping multiple vehicle properties into a small set of permissions, and assigns inadequate protection levels, reducing meaningful user consent [109]. To address this, OEMs must provide clear sensor-level explanations, and third-party apps should justify each requested permission to prevent over-privileged access.

Data Anonymization: When sharing the data with the third-party app provider, the privacy goal of data anonymization will eliminate any personally identifying information (PII) from the data. This can be achieved by modifying data so that it is no longer possible to identify the data subject. Pseudonymization, generalization, data masking, swapping, perturbation, and synthetic data generation are the six categories of data anonymization techniques [128].

Data Sanitization: Sharing reduced or anonymized data with third parties may still violate GDPR principles of purpose limitation and storage limitation. Once data are transferred, the OEM loses practical control over third-party use and retention, even if constrained by the Terms of Service (ToS). As shown in Fig. 11b, an *honest-but-curious* UBI provider authorized to collect braking, steering angle, and acceleration data for scoring can still perform driver fingerprinting and infer undisclosed drivers. Although illegal, such misuse cannot be technically prevented post-sharing. ISO/SAE 21434 similarly highlights that road user data are highly sensitive and easily linkable to PII [76].

5.2.2 *Attacks.*

A survey on inference attacks demonstrated how seemingly benign vehicular data can reveal more context than initially apparent [113]. These inference attacks violate the GDPR Principle of Purpose Limitation and are categorized into two types: (1) *Driver Fingerprinting*, (2) *Location Inference* [113]. To assess the risk level of these attacks, a metric called Privacy Score (PS) was defined for 20 frequently collected vehicular sensors, indicating the potential for inferring PII. The analysis identified location, vehicle speed, and steering wheel angle as the three most privacy-sensitive sensors.

Driver Fingerprinting. Driver fingerprinting aims to identify individuals operating the same vehicle using sensor data collected during trips. Prior work showed that combining 15 sensors can identify 15 drivers with 100% accuracy [48]. Another study demonstrated that naturalistic driving behavior enables driver identification within minutes by training a random forest model on several hours of data [155]. Further analysis revealed that fuel trim, brake pedal, and steering wheel data support accurate re-identification with only a few minutes of collected data using machine learning models [50]. Additional work showed that drivers can be identified even before trip start, achieving high precision shortly after vehicle entry [79]. More recently, driver fingerprinting was extended to raw CAN data, reaching 97% re-identification accuracy [55].

Location Inference. Vehicle geolocation, one of the most privacy-sensitive in-vehicle data types, can be inferred from less invasive sensors such as acceleration, speed, and steering wheel angle [110]. Early work showed that mobile Inertial Measurement Unit (IMU) data enable path inference using Depth-First Search (DFS) [43]. With On-Board Diagnostics-II (OBD-II) insurance dongles, Elastic Pathing enabled destination prediction from speed data [54]. Subsequent studies improved route identification using DFS and Hidden Markov Models [163, 164]. Map-matching based on distances and turning directions enabled large-scale inference without prior location knowledge [154]. Steering wheel angle [110] and brake signal data [127] further supported route reconstruction. Recent attacks combined CAN and OBD-II data with physical vehicle models for accurate path reconstruction [17], with meter-level trajectory errors over short distances [55].

5.2.3 *Defenses.*

Although regulation such as GDPR or ISO/SAE 21434 can be regarded as a non-technical defense, the following subsection discusses technical defenses against aforementioned privacy attacks in academic literature.

Differential privacy was introduced to the automotive industry as a method to mitigate privacy risks associated with vehicle data collection, providing protection for driver privacy while allowing data analysis [104]. This approach also addressed challenges in applying differential privacy to multidimensional time series data and proposed solutions such as personalized privacy budgets, random sampling, and event-level privacy to balance utility and privacy [104]. An evaluation of anonymization techniques, including smoothing, low-pass filtering, and aggregation, demonstrated their effectiveness in reducing re-identification accuracy and increasing trajectory reconstruction errors for driver fingerprinting and location inference attacks [55].

The PRICAR framework was proposed to enable privacy-preserving collection and sharing of vehicle data with third parties [111]. PRICAR enforces data minimization, anonymization, and sanitization, with particular emphasis on sanitization techniques. It relies on a neutral entity that executes third-party code within a sandboxed environment, preventing direct data access and allowing OEMs to sanitize data before returning results. This design mitigates privacy risks and supports compliance with regulations such as GDPR.

Another privacy-aware data access system designed for automotive applications was developed to comply with GDPR requirements [115]. This system informs users about sensitive data flows and provides control over third-party access to personal data. Its human-machine interface (HMI)

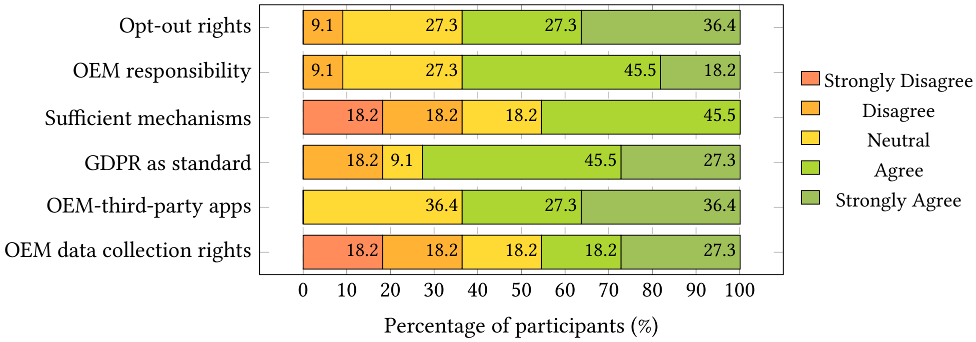


Fig. 12. Experts' answers on privacy and data collection for SDV, where a rating of 1 represents strong disagreement and 5 indicates strong agreement.

and policy framework enable users to define and manage their privacy settings while ensuring transparency and control over data usage.

5.2.4 ***Experts' Feedback*** As shown in Fig. 12, expert responses on privacy exhibit varying degrees of consensus. The strongest agreement concerns *Opt-Out Rights for Data Collection*, with a median of 4, low dispersion (IQR = 1), and high endorsement (TTBA ≈ 64%), indicating broad support for allowing drivers to opt out of data collection while retaining infotainment access. A similar pattern is observed for *OEM Responsibility for Data Sanitization* (IQR = 1, TTBA ≈ 64%), reflecting the view that OEMs, as data controllers under the GDPR, bear primary responsibility for sanitizing shared data. By contrast, opinions on the *Sufficiency of Existing Privacy Mechanisms* are more divided (IQR = 2, TTBA ≈ 46%), indicating disagreement on whether current infotainment privacy protections are adequate. Strong consensus is again visible for *GDPR as the Privacy Standard* (IQR = 1, TTBA > 70%), confirming its role as the dominant reference framework for privacy design in SDVs. Responses related to *OEM Control over Third-Party Apps* and *OEM Data Collection Rights* show higher dispersion (IQR = 2), highlighting ongoing debate on governance and data control. Overall, experts align on user rights and regulatory responsibility, while governance mechanisms remain more contested.

Takeaway 9. Privacy concerns should be addressed with a focus on transparency, user consent and data governance. With SDVs collecting vast amounts of sensitive data, including geolocation, driver behavior, and even personal identifiers, maintaining user trust is crucial. Transparent data collection practices and clear user consent mechanisms must be integrated to address privacy concerns effectively. Providing users with control over their data can increase trust and compliance with privacy regulations while also fostering positive user engagement. A key challenge lies in balancing data utility and privacy guarantees, a trade-off that is particularly critical in SDVs, where data is central to both system operation and business models. Privacy in SDVs must account for time-evolving data access, where OTA updates can introduce new data collection capabilities post-deployment.

6 Related work and comparison

Our review shows that many works retrieved using the term "software-defined vehicle" actually address adjacent concepts, such as software-defined vehicular networking or Internet of Vehicles, rather than SDVs as a distinct paradigm. As a result, only a limited number of surveys explicitly focus on SDVs, and existing definitions remain fragmented. In the following, we discuss the most relevant related works and position our contribution accordingly. A structured comparison is provided in Table 3.

SDV-focused security surveys. To the best of our knowledge, the first academic review explicitly addressing SDVs security is the 2023 work by Bodei *et al.* [20]. That study investigates the transition from hardware-centric vehicles to SDVs, proposes an initial definition, and identifies emerging vulnerabilities using ISO/SAE 21434, including DoS and jamming attacks. It highlights the limited academic attention to SDV-specific security despite strong industrial interest. Our work extends this contribution by refining the SDV definition, expanding the analyzed attack surfaces, and incorporating OTA, privacy, and expert-driven insights. A closely related survey is presented by Sghaier *et al.* [132], who provide a valuable contribution by proposing a comprehensive taxonomy and threat model that map cyberattacks to SDV architectural properties. Their work offers a detailed classification of vulnerabilities across in-vehicle, connectivity, and cloud components, helping to structure SDV-specific security research. Our work adopts a broader systematization-of-knowledge perspective, providing an integrated treatment of security, OTA, and privacy across the SDV lifecycle. Moreover, it incorporates an industry-oriented viewpoint grounded in expert elicitation and multidisciplinary academic and industrial expertise of the authors, and explicitly proposes mitigation strategies, an aspect not addressed in previous surveys.

SDV definition and service-oriented perspectives. Teixeira *et al.* [143] examine SDVs from a deterministic service-development perspective, emphasizing predictable software behavior in complex vehicles. More recently, Mate *et al.* [96] bridge industrial and academic perspectives on SDVs, by discussing architectural evolution, development practices, and selected security aspects. While their work acknowledges security challenges, it does not provide a systematic threat analysis nor addresses OTA, privacy, or lifecycle risk management in depth. In contrast, our survey integrates these dimensions within a unified framework supported by expert elicitation.

Architectural and system-level SDV discussions. El-Fatyany *et al.* [46] analyze architectural, control, and security challenges arising from software-centric vehicle designs, emphasizing the increased attack surface introduced by connectivity and centralized computation. Although aligned with several of our findings, their study does not systematically compare SDVs with other vehicle paradigms nor provides an integrated analysis of OTA or privacy aspects.

Broader automotive software and SaaS surveys. Several surveys address automotive software security without explicitly targeting SDVs. For example, Huynh Le *et al.* [86] present a systematic survey of security and privacy in automotive software platforms, but predate the SDV paradigm and therefore do not capture its architectural and lifecycle-specific challenges. Similarly, Blanco *et al.* [18] analyze the transformation of automotive systems toward Software as a Service (SaaS), focusing on architecture, software pipelines, and runtime management. While relevant for understanding software-driven vehicle evolution, this work is not security-centric and does not address threat models or defensive mechanisms, serving instead as complementary background.

Positioning of our contribution. Overall, existing surveys either address vehicular cybersecurity broadly, focus on software and architectural transformation without a dedicated security lens, or propose SDV-specific taxonomies without integrating expert prioritization, privacy, and OTA-centric lifecycle analysis. This work differentiates itself by offering a holistic SDV-focused perspective that jointly considers security, OTA, and privacy, explicitly contrasts SDVs with AVs and CVs, and incorporates expert elicitation to bridge academic analysis with industrial practice.

7 Conclusion

This survey provides a comprehensive overview of cybersecurity and privacy challenges in SDVs, identifying key attack surfaces, vulnerabilities, mitigation strategies, and privacy risks. The analysis highlights the need for security frameworks capable of adapting to the software-centric evolution.

Table 3. Related work comparison.

Article	Year	Definition SDV	Comparison SDV/AV/CV	Attacks	Defenses	OTA	Privacy	Industry perspective
Huynh Le *et al.* [86]	2018	❍	❍	●	●	❍	●	❍
Bodei *et al.* [20]	2023	●	❍	●	●	❍	❍	❍
Blanco *et al.* [18]	2023	●	❍	❍	❍	●	❍	❍
Teixeira *et al.* [143]	2024	●	❍	❍	❍	❍	❍	❍
El-Fatyany *et al.* [46]	2024	●	❍	●	●	❍	❍	❍
Mate *et al.* [96]	2025	●	●	●	❍	❍	❍	❍
Sghaier *et al.* [132]	2025	●	❍	●	●	●	●	❍
Our work	2026	●	●	●	●	●	●	●

Takeaway (Final Synthesis). Security and privacy in SDVs must be understood as an interconnected, system-level problem spanning attack surfaces, mitigations, OTA mechanisms, and data governance. As highlighted in this work, specific attack surfaces, such as APIs, third-party software, supply chains, and OTA channels, are tightly coupled with continuous software evolution and distributed architectures. These surfaces cannot be addressed in isolation: vulnerabilities in one layer (e.g., insecure APIs or supply-chain components) can propagate through OTA updates and impact multiple vehicle functions at scale.

Mitigation strategies must therefore be coordinated across the lifecycle, combining secure software development, intrusion detection, and update verification with privacy-preserving mechanisms. Overall, ensuring trustworthy SDVs requires a holistic approach that jointly considers security and privacy across all layers of the ecosystem, from in-vehicle components to cloud services and multi-stakeholder environments.

Future work. Several research directions remain critical for strengthening SDV security and privacy. Intrusion detection systems tailored to SDVs must evolve toward adaptive, AI-driven approaches capable of detecting complex attacks across in-vehicle networks, APIs, and OTA processes. In parallel, supply chain security requires end-to-end frameworks that address risks arising from third-party software and hardware across the SDV lifecycle. Given the extensive collection of sensitive data, privacy-preserving mechanisms such as differential privacy remain essential to protect PII while enabling data-driven services. OTA security represents another key area in which decentralized and edge-assisted approaches may improve integrity, resilience, and transparency of update processes. As APIs become foundational to SDVs, interface standardization is necessary to reduce misconfigurations and exposure to exploitation. In this context, emerging AI-based solutions, including Large Language Models (LLMs) and Vision-Language Models (VLMs), offer new opportunities within ITS for security monitoring, anomaly detection, and policy reasoning, while also introducing novel attack surfaces that require careful governance, in particular in future classes of vehicles such as AI-defined Vehicles (AIDV) [38]. Continued alignment with standards such as ISO/SAE 21434 and UNECE WP.29 will be essential to ensure that security and privacy frameworks evolve alongside these technologies, ultimately shaping the trustworthiness of future SDVs.

Acknowledgments

We would like to thank all of the experts in the automotive industry who generously contributed their time and expertise by participating in the survey. Their valuable insights and feedback have been crucial in shaping the findings of this research. This work has been partially supported by the PNRR project Securing sOftware Platform (SOP), by the program PTR 22-24 P2.01 (Cybersecurity) and SERICS (PE00000014) under the NRRP MUR program funded by the EU, and the US National Science Foundation Award 2345653. In addition, the work was supported by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder in the context of the German-French Academy for the Industry of the Future of Institut Mines-Télécom (IMT) and the Technical University of Munich (TUM). Any findings, opinions, recommendations or conclusions expressed in the article are those of the authors and do not necessarily reflect the views of the sponsor.

References

[1] Mohammad Y. Abualhoul, Oyunchimeg Shagdar, and Fawzi Nashashibi. 2016. Visible Light inter-vehicle Communication for platooning of autonomous vehicles. In *2016 IEEE Intelligent Vehicles Symposium (IV)*. 508–513. doi:10.1109/IVS.2016.7535434

[2] Accenture. 2022. Moving into the Software-Defined Vehicle Fast Lane. https://www.accenture.com/content/dam/accenture/final/industry/mobility/document/Accenture-Software-Defined-Vehicles-pov.pdf, Last accessed March 26, 2026.

[3] ACM. 2024. *ACM Digital Library.* https://dl.acm.org/ Last accessed March 26, 2026.

[4] Emad Aliwa, Omer Rana, Charith Perera, and Peter Burnap. 2021. Cyberattacks and Countermeasures for In-Vehicle Networks. *ACM Comput. Surv.* 54, 1, Article 21 (mar 2021), 37 pages. doi:10.1145/3431233

[5] Allstate. 2024. Drivewise - Allstate. https://www.allstate.com/drive-wise/drivewise-device.aspx.

[6] Allstate. 2024. Esurance Insurance Company. https://www.esurance.com/drivesense.

[7] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2020. Dos and Don'ts of Machine Learning in Computer Security. *CoRR* abs/2010.09470 (2020). arXiv:2010.09470 https://arxiv.org/abs/2010.09470

[8] Nadarajah Asokan, Thomas Nyman, Norrathep Rattanavipanon, Ahmad-Reza Sadeghi, and Gene Tsudik. 2018. ASSURED: Architecture for secure software update of realistic embedded devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 11 (2018), 2290–2300.

[9] AUTOSAR Consortium. 2024. *AUTOSAR Adaptive Platform.* AUTOSAR. https://www.autosar.org/standards/adaptive-platform/ Last accessed March 26, 2026.

[10] AUTOSAR Partnership. 2023. AUTOSAR Opening Strategy for the Software-Defined Vehicle Ecosystem. https://www.autosar.org/news-events/detail/autosar-opening-strategy-software-defined-vehicle-ecosystem. Last accessed March 26, 2026.

[11] Omid Avatefipour and Hafiz Malik. 2018. State-of-the-Art Survey on In-Vehicle Network Communication (CAN-Bus) Security and Vulnerabilities. arXiv:1802.01725 [cs.CR] https://arxiv.org/abs/1802.01725

[12] Jeanseong Baik, Haegeon Jeong, and Kyungtae Kang. 2020. Poster: Network-Centric Approach Using Task Migration for Drive-by-Wire Vehicle Resilience. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. IEEE, 1–2.

[13] BBC. 2020. *Trier: Five die as car ploughs through Germany pedestrian zone.* https://www.bbc.com/news/world-europe-55148518 Last accessed March 26, 2026.

[14] Jan Becker. 2022. A Safety-Certified Automotive SDK to Enable Software-Defined Vehicles. In *Workshop Fahrerassistenz und automatisiertes Fahren.* https://www.uni-das.de/images/pdf/fas-workshop/2022/FAS2022-12-Becker.pdf

[15] Maite Bezerra. 2025. *AI Takes Center Stage in Software-Defined Vehicle Evolution.* https://www.wardsauto.com/news/ai-takes-center-stage-in-software-defined-vehicle-evolution/798938/ Last accessed March 26, 2026.

[16] Arpan Bhattacharjee, Hamza Mahmood, Sidi Lu, Nejib Ammar, Akila Ganlath, and Weisong Shi. 2023. Edge-Assisted Over-the-Air Software Updates. In *2023 IEEE 9th International Conference on Collaboration and Internet Computing (CIC)*. IEEE Computer Society, Los Alamitos, CA, USA, 18–27. doi:10.1109/CIC58953.2023.00013

[17] Tommaso Bianchi, Alessandro Brighente, Mauro Conti, and Andrea Valori. 2024. Your Car Tells Me Where You Drove: A Novel Path Inference Attack via CAN Bus and OBD-II Data. *arXiv preprint arXiv:2407.00585* (2024).

[18] David Fernández Blanco, Frédéric Le Mouël, Trista Lin, and Marie-Pierre Escudié. 2023. A Comprehensive Survey on Software as a Service (SaaS) Transformation for the Automotive Systems. *IEEE Access* 11 (2023), 73688–73753. doi:10.1109/ACCESS.2023.3294256

[19] Chiara Bodei, Marco De Vincenzi, and Ilaria Matteucci. 2024. Formal Analysis of an AUTOSAR-Based Basic Software Module. 26, 4 (2024), 495–508. doi:10.1007/s10009-024-00759-w

[20] Chiara Bodei, Marco De Vincenzi, and Ilaria Matteucci. 2023. From Hardware-Functional to Software-Defined Vehicles and their Security Issues. In *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. 1–10. doi:10.1109/INDIN51400.2023.10217971

[21] Molly Boigon. 2024. *Software-defined vehicles are all the rage. Too bad they don't exist yet.* https://www.autonews.com/mobility-report/software-defined-vehicles-will-require-supply-chain-and-revenue-strategy-shifts https://www.autonews.com/mobility-report/software-defined-vehicles-will-require-supply-chain-and-revenue-strategy-shifts, Last accessed March 26, 2026.

[22] Nick Bondaug-Winn. 2023. *Understanding the Impact of Autonomous Vehicles on Insurance Agencies.* https://www.hbwleads.com/blog/understanding-the-impact-of-autonomous-vehicles-on-insurance-agencies/, Last accessed March 26, 2026.

[23] Andrew Booth, Anthea Sutton, and Diana Papaioannou. 2016. *Systematic Approaches to a Successful Literature Review.* SAGE Publications. https://books.google.com/books?id=JD1DCgAAQBAJ

[24] Bosch. 2023. Bosch software-defined vehicle. https://www.bosch-mobility.com/en/mobility-topics/software-defined-vehicle/. Last accessed March 26, 2026.

[25] Christoph Bösch, Benjamin Erb, Frank Kargl, Henning Kopp, and Stefan Pfattheicher. 2016. Tales from the dark side: Privacy dark strategies and privacy dark patterns. *Proceedings on Privacy Enhancing Technologies* (2016).

[26] Siham Bouchelaghem, Abdelmadjid Bouabdallah, and Mawloud Omar. 2021. *Autonomous Vehicle Security: Literature Review of Real Attack Experiments.* 255–272. doi:10.1007/978-3-030-68887-5_15

[27] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. 2021. Invisible for both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks. In *2021 IEEE Symposium on Security and Privacy (SP).* 176–194. doi:10.1109/SP40001.2021.00076

[28] Miller Charlie and Valasek Chris. 2015. Remote Exploitation of an Unaltered Passenger Vehicle. https://illmatics.com/Remote%20Car%20Hacking.pdf Last accessed March 26, 2026.

[29] Abigail R Colson and Roger M Cooke. 2018. *Expert elicitation: using the classical model to validate experts' judgments.* The University of Chicago Press.

[30] Cookiebot. 2020. California Privacy Rights Act (CPRA): CCPA VS CPRA. https://www.cookiebot.com/en/cpra/ Last accessed March 26, 2026.

[31] Renesas Electronics Corporation. 2024. *The Art of Networking Series 9: SDN - The Next Hype after Automotive Ethernet?* https://www.renesas.com/en/blogs/art-networking-series-9-sdn-next-hype-after-automotive-ethernet, Last accessed March 26, 2026.

[32] Xiaomi Corporation. 2025. Xiaomi Automotive: Electric Vehicles and Smart Car Technologies. https://www.xiaomiev.com/.

[33] Gianpiero Costantino, Marco De Vincenzi, and Ilaria Matteucci. 2024. A vehicle firmware security vulnerability: an IVI exploitation. *J. Comput. Virol. Hacking Tech.* 20, 4 (2024), 681–696. doi:10.1007/S11416-024-00522-4

[34] Sam Curry. 2024. *Hacking Kia: Remotely Controlling Cars With Just a License Plate.* https://samcurry.net/hacking-kia, Last accessed March 26, 2026.

[35] Cybersecurity and Infrastructure Security Agency (CISA). 2022. Vehicle Ramming: Security Awareness for Soft Targets and Crowded Places. https://www.cisa.gov, Last accessed March 26, 2026.

[36] National Vulnerability Database. 2023. CVE-2023-1709: Vulnerability in [Vulnerable Product/Component]. https://nvd.nist.gov/vuln/detail/CVE-2023-1709 Last accessed March 26, 2026.

[37] Marco De Vincenzi, Chiara Bodei, and Ilaria Matteucci. 2023. Securing Automotive Ethernet: Design and Implementation of Security Data Link Solutions. In *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA).* 1–9. doi:10.1109/AICCSA59173.2023.10479353

[38] Marco De Vincenzi, Chiara Bodei, and Ilaria Matteucci. 2026. When AI takes the wheel: AI-defined vehicles principles and pitfalls. *Frontiers in Robotics and AI* Volume 13 - 2026 (2026). doi:10.3389/frobt.2026.1770121

[39] Marco De Vincenzi, Gianpiero Costantino, Ilaria Matteucci, Florian Fenzl, Christian Plappert, Roland Rieke, and Daniel Zelle. 2024. A Systematic Review on Security Attacks and Countermeasures in Automotive Ethernet. *ACM Comput. Surv.* 56, 6, Article 135 (Jan. 2024), 38 pages. doi:10.1145/3637059

[40] Marco De Vincenzi, John Moore, Bradley Smith, Sanjay E. Sarma, and Ilaria Matteucci. 2025. Security Risks and Designs in the Connected Vehicle Ecosystem: In-Vehicle and Edge Platforms. *IEEE Open Journal of Vehicular Technology* 6 (2025), 442–454. doi:10.1109/OJVT.2024.3524088

[41] Deloitte. 2023. Software-defined Vehicles. https://www2.deloitte.com/us/en/pages/consumer-business/articles/the-software-defined-vehicle-revolution.html. Last accessed March 26, 2026.

[42] GM Developers. 2024. GM Developers. https://developer.gm.com/in-vehicle-apps Last accessed March 26, 2026.

[43] Rinku Dewri, Prasad Annadata, Wisam Eltarjaman, and Ramakrishna Thurimella. 2013. Inferring trip destinations from driving habits data. In *Proc. of the 12th ACM workshop on Workshop on privacy in the electronic society.* 267–272.

[44] Mark Dolan. 2025. *Issue 272: Volkswagen API hacked, API flaws in Instagram & TikTok, ELi attacks, Radware & Cisco API vulnerabilities.* https://apisecurity.io/issue-272-volkswagen-api-hacked-api-flaws-in-instagram-tiktok-eli-attacks-radware-cisco-api-vulnerabilities/ Last accessed March 26, 2026.

[45] Eaton. 2023. Toyota C360 Hack. https://eaton-works.com/2023/03/06/toyota-c360-hack/. Last accessed March 26, 2026.

[46] Aya El-Fatyany, Xiaohang Wang, Parasara Duggirala, Samarjit Chakraborty, Sudeep Pasricha, and Amit Singh. 2024. Special Session: Emerging Architecture Design, Control, and Security Challenges in Software Defined Vehicles.

[47] William Enck, Damien Octeau, Patrick D McDaniel, and Swarat Chaudhuri. 2011. A study of android application security. In *USENIX security symposium*, Vol. 2.

[48] Miro Enev, Alex Takakuwa, Karl Koscher, and Tadayoshi Kohno. 2016. Automobile driver fingerprinting. *Proceedings on Privacy Enhancing Technologies* (2016).

[49] European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). http://data.europa.eu/eli/reg/2016/679/oj.

[50] Saad Ezzini, Ismail Berrada, and Mounir Ghogho. 2018. Who is behind the wheel? Driver identification and fingerprinting. *Journal of Big Data* 5, 1 (2018), 1–15.

[51] State Farm. 2018. Drive Safe & Save™ – State Farm®. https://www.statefarm.com/insurance/auto/discounts/drive-safe-save, Last accessed March 26, 2026.

[52] Ford. 2024. Ford Developer Marketplace. https://developer.ford.com/infotainment/in-vehicle-downloadable-apps Last accessed March 26, 2026.

[53] Mozilla Foundation. 2023. 'Privacy Nightmare on Wheels'. https://foundation.mozilla.org/en/blog/privacy-nightmare-on-wheels-every-car-brand-reviewed-by-mozilla-including-ford-volkswagen-and-toyota-flunks-privacy-test Last accessed March 26, 2026.

[54] Xianyi Gao, Bernhard Firner, Shridatt Sugrim, Victor Kaiser-Pendergrast, Yulong Yang, and Janne Lindqvist. 2014. Elastic pathing: Your speed is enough to track you. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. 975–986.

[55] András Gazdag, Szilvia Lestyán, Mina Remeli, Gergely Ács, Tamás Holczer, and Gergely Biczók. 2023. Privacy pitfalls of releasing in-vehicle network data. *Vehicular Communications* 39 (2023), 100565.

[56] Amrita Ghosal and Mauro Conti. 2020. Security issues and challenges in V2X: A survey. *Computer Networks* 169 (2020), 107093.

[57] GlobeNewsWire. 2019. Daimler Partners with Otonomo to Provide Connected Car Customers with New Services while Delivering on the Promise of Data Privacy. https://www.globenewswire.com/news-release/2019/01/10/1685883/0/en/UPDATED-Daimler-Partners-with-Otonomo-to-Provide-Connected-Car-Customers-with-New-Services-while-Delivering-on-the-Promise-of-Data-Privacy.html Last accessed March 26, 2026.

[58] Google Scholar. 2024. *Google Scholar*. https://scholar.google.com/ Last accessed March 26, 2026.

[59] Colin M Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L Toombs. 2018. The dark (patterns) side of UX design. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–14.

[60] BMW Group. 2017. BMW Group launches BMW CarData: new and innovative services for customers, safely and transparently. https://www.press.bmwgroup.com/global/article/detail/T0271366EN/bmw-group-launches-bmw-cardata:-new-and-innovative-services-for-customers-safely-and-transparently?language=en Last accessed March 26, 2026.

[61] Ujjwal Guin, Ke Huang, Daniel DiMase, John M Carulli, Mohammad Tehranipoor, and Yiorgos Makris. 2014. Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain. *Proc. IEEE* 102, 8 (2014), 1207–1228.

[62] Subir Halder, Amrita Ghosal, and Mauro Conti. 2020. Secure over-the-air software updates in connected vehicles: A survey. *Computer Networks* 178 (2020), 107343.

[63] Mohammad Hamad, Zain AH Hammadeh, Davide Alessi, Monowar Hasan, Mert Pese, Daniel Lüdtke, and Sebastian Steinhorst. 2025. Enhancing Security Through Task Migration in Software-Defined Vehicles. *IEEE Internet of Things Journal* (2025).

[64] Mohammad Hamad and Vassilis Prevelakis. 2017. Secure APIs for Applications in Microkernel-based Systems. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP*. INSTICC, SciTePress, 553–558. doi:10.5220/0006265805530558

[65] Mohammad Hamad and Vassilis Prevelakis. 2020. SAVTA: A hybrid vehicular threat model: Overview and case study. *Information* 11, 5 (2020), 273.

[66] Mona Helmy and Mohamed Mahmoud. 2023. Enhanced Multi-Level Secure Over-the-Air Update System using Adaptive AUTOSAR. In *2023 International Conference on Computer and Applications (ICCA)*. 1–4. doi:10.1109/ICCA59364.2023.10401797

[67] Kashmir Hill. 2024. How G.M. Tricked Millions of Drivers Into Being Spied On (Including Me). *N.Y. Times* (April 2024). https://www.nytimes.com/2024/04/23/technology/general-motors-spying-driver-data-consent.html

[68] Honda. 2024. Honda Vehicles with Google built-in. https://automobiles.honda.com/google-built-in Last accessed March 26, 2026.

[69] Lois Hoyal. 2024. Automakers forecast to earn tenfold more revenue from digital services. https://europe.autonews.com/automakers/why-software-defined-vehicles-offer-big-profit-potential, Last accessed March 26, 2026.

[70] Jeremy Hsu. 2014. Toyota recalls 1.9 million prius hybrids over software flaw. *IEEE Spectrum, Feb* (2014).

[71] IBM. 2023. The Software Defined Vehicle - IBM Blog - Digitale Perspektive. https://www.ibm.com/blogs/digitale-perspektive/2023/06/the-software-defined-vehicle/. Last accessed March 26, 2026.

[72] IDTechEx. 2025. Software-Defined Vehicles, Connected Cars, and AI in Cars: Market Trends and Business Models. https://www.idtechex.com/en/research-report/software-defined-vehicles-connected-cars-and-ai-in-cars/1108. Last accessed March 26, 2026.

[73] IEEEXplore. 2024. *IEEE Xplore Library.* https://ieeexplore.ieee.org/Xplore/home.jsp Last accessed March 26, 2026.

[74] SAE International. 2021. SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. https://www.sae.org/standards/content/j3016

[75] ISO. 2021. *Road vehicles — Functional safety.* Standard ISO 26262:2018. International Organization for Standardization, Geneva, CH. https://www.iso.org/standard/68383.html

[76] ISO. 2021. *Road vehicles — Cybersecurity engineering.* Standard ISO/SAE FDIS 21434:2021 Ed.1. International Organization for Standardization, Geneva, CH. https://www.iso.org/standard/70918.html

[77] Japan Today. 2019. *8 injured as man rams car into pedestrians in Harajuku in 'retaliation for execution'.* https://japantoday.com/category/crime/8-injured-as-man-rams-car-into-pedestrians-in-Harajuku-in-%27retaliation-for-execution%27 Last accessed March 26, 2026.

[78] Boosun Jeon, Hongil Ju, Boheung Jung, Kyungtae Kim, and Duyeon Lee. 2019. A Study on Traffic Characteristics for Anomaly Detection of Ethernet-based IVN. In *2019 International Conference on Information and Communication Technology Convergence (ICTC).* 951–953. doi:10.1109/ICTC46691.2019.8940022

[79] Gorkem Kar, Shubham Jain, Marco Gruteser, Jinzhu Chen, Fan Bai, and Ramesh Govindan. 2017. PredriveID: Pre-trip driver identification from in-vehicle data. In *Proc. of the Second ACM/IEEE Symposium on Edge Computing.* 1–12.

[80] Pearse Keane. 2024. *The Software-Defined Vehicle: Impacts Across the Automotive Ecosystem.* Jabil. https://www.jabil.com/blog/software-defined-vehicle.html, Last accessed March 26, 2026.

[81] Casper Kessels. 2024. The state of Android Automotive in 2024 - Snapp Automotive. https://www.snappautomotive.io/blog/the-state-of-android-automotive-in-2024 Last accessed March 26, 2026.

[82] Patrick Kingsley, Euan Ward, Ronen Bergman, and Michael Levenson. 2024. *Exploding Pagers Targeting Hezbollah Kill 11 and Wound Thousands.* https://www.nytimes.com/2024/09/17/world/middleeast/hezbollah-pager-explosions-lebanon.html, Last accessed March 26, 2026.

[83] Piergiorgio Ladisa, Henrik Plate, Matias Martinez, and Olivier Barais. 2023. SoK: Taxonomy of Attacks on Open-Source Software Supply Chains. In *2023 IEEE Symposium on Security and Privacy (SP).* 1509–1526. doi:10.1109/SP46215.2023.10179304

[84] Brooke Lampe and Weizhi Meng. 2023. Intrusion Detection in the Automotive Domain: A Comprehensive Review. *IEEE Communications Surveys & Tutorials* 25, 4 (2023), 2356–2426. doi:10.1109/COMST.2023.3309864

[85] Aljoscha Lautenbach, Magnus Almgren, and Tomas Olovsson. 2021. Proposing HEAVENS 2.0–an automotive risk assessment model. In *Proceedings of the 5th ACM Computer Science in Cars Symposium.* 1–12.

[86] Van Huynh Le, Jerry den Hartog, and Nicola Zannone. 2018. Security and privacy for innovative automotive applications: A survey. *Computer Communications* 132 (2018), 17–41. doi:10.1016/j.comcom.2018.09.010

[87] Namcheol Lee, Seongsoo Hong, and Saehwa Kim. 2024. Dynamic Mapping of Mixed-Criticality Applications onto a Mixed-Criticality Runtime System with Probabilistic Guarantees. In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS).* IEEE, 1466–1467.

[88] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *2007 IEEE 23rd International Conference on Data Engineering.* 106–115. doi:10.1109/ICDE.2007.367856

[89] Rensis Likert. 1932. A Technique for the Measurement of Attitudes. *Archives of Psychology* 22, 140 (1932), 1–55.

[90] Zongwei Liu, Wang Zhang, and Fuquan Zhao. 2018. Security and privacy for innovative automotive applications: A survey. *Computer Communications* 5 (2018), 17–41. doi:10.1016/j.comcom.2018.09.010

[91] Zongwei Liu, Wang Zhang, and Fuquan Zhao. 2022. Impact, Challenges and Prospect of Software-Defined Vehicles. *Automotive Innovation* 5 (2022), 180–194. doi:10.1007/s42154-022-00179-z

[92] Waymo LLC. 2025. Waymo: Self-Driving Technology for Autonomous Vehicles. https://waymo.com/.

[93] Aaron Luo and Vit Sembera. 2023. *Two Tesla Hacks Triumph at Pwn2Own Vancouver 2023.* VicOne Inc. https://vicone.com/blog/two-tesla-hacks-triumph-at-pwn2own-2023 Last accessed March 26, 2026.

[94] Asad Waqar Malik, Anis U. Rahman, Arsalan Ahmad, and Max Mauro Dias Santos. 2022. Over-the-Air Software-Defined Vehicle Updates Using Federated Fog Environment. *IEEE Transactions on Network and Service Management* 19, 4 (2022), 5078–5089. doi:10.1109/TNSM.2022.3181027

[95] Stefan F Marksteiner, Christoph Schmittner, Korbinian Christl, Dejan Nickovic, Mikael Sjödin, and Marjan Sirjani. 2023. From TARA to Test: Automated Automotive Cybersecurity Test Generation Out of Threat Modeling. In *Proceedings of the 7th ACM Computer Science in Cars Symposium* (Darmstadt, Germany) *(CSCS '23)*. Association for Computing Machinery, New York, NY, USA, Article 5, 10 pages. doi:10.1145/3631204.3631864

[96] Balint Mate, Max Scheerer, Ralf Sieger, Oliver Denninger, and Jorg Henss. 2025. Software-Defined Vehicles: Bridging Industry and Research Perspectives. *IEEE International Symposium on Industrial Electronics* (2025). doi:10.1109/ISIE62713.2025.11124637

[97] Matt McFarland, CNN business. 2022. Teen's Tesla hack shows how vulnerable third-party apps may make cars. https://www.cnn.com/2022/02/02/cars/tesla-teen-hack/index.html. Last accessed March 26, 2026.

[98] Rani Molla. 2024. How to opt out of the privacy nightmare that comes with new Hondas. *Sherwood News* (May 2024). https://sherwood.news/tech/how-to-opt-out-of-the-privacy-nightmare-that-comes-factory-installed-in-new

[99] Jean-Philippe Monteuuis, Aymen Boudguiga, Jun Zhang, Houda Labiod, Alain Servel, and Pascal Urien. 2018. Sara: Security automotive risk analysis method. In *Proc. of the 4th ACM Workshop on Cyber-Physical System Security*. 3–14.

[100] Marina Moore, Aditya Sirish A Yelgundhalli, Trishank Karthik Kuppusamy, Santiago Torres-Arias, Lois Anne DeLong, and Justin Cappos. 2022. *Scudo: A Proposal for Resolving Software Supply Chain Insecurities in Vehicles.* Technical Report. Uptane Standards Group. https://uptane.org/papers/scudo-whitepaper.pdf Whitepaper; released May 2022, updated July 2022.

[101] Lama J. Moukahal, Mohammad Zulkernine, and Martin Soukup. 2021. Towards a Secure Software Lifecycle for Autonomous Vehicles. In *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 371–377. doi:10.1109/ISSREW53611.2021.00104

[102] National Highway Traffic Safety Administration (NHTSA). 2023. Automated Vehicles for Safety | NHTSA. https://www.nhtsa.gov/vehicle-safety/automated-vehicles-safety. Last accessed March 26, 2026.

[103] National Institute of Standards and Technology. 2024. The NIST Cybersecurity Framework (CSF) 2.0. https://www.nist.gov/cyberframework Last accessed March 26, 2026.

[104] Boel Nelson and Tomas Olovsson. 2017. Introducing Differential Privacy to the Automotive Domain: Opportunities and Challenges. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. 1–7. doi:10.1109/VTCFall.2017.8288389

[105] Governor of California. 2018. Basics of the California Consumer Privacy Act of 2018. https://www.privacypolicies.com/blog/california-consumer-privacy-act/.

[106] Dennis Kengo Oka. 2021. *Building secure cars: assuring the automotive software development lifecycle.* John Wiley & Sons.

[107] Akash Pallath and Qiyang Zhang. 2023. Paperfetcher: A tool to automate handsearching and citation searching for systematic reviews. *Res. Synth. Methods* 14, 2 (March 2023), 323–335. https://doi.org/10.1002/jrsm.1604

[108] Fengjunjie Pan, Jianjie Lin, Markus Rickert, and Alois Knoll. 2022. Resource Allocation in Software-Defined Vehicles: ILP Model Formulation and Solver Evaluation. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. 2577–2584. doi:10.1109/ITSC55140.2022.9922526

[109] Mert D Pese. 2023. *A first look at android automotive privacy.* Technical Report. SAE Technical Paper.

[110] Mert D Pesé, Xiaoying Pu, and Kang G Shin. 2020. Spy: Car steering reveals your trip route! *Proceedings on Privacy Enhancing Technologies* (2020).

[111] Mert D Pesé, Jay W Schauer, Murali Mohan, Cassandra Joseph, Kang G Shin, and John Moore. 2023. PRICAR: Privacy Framework for Vehicular Data Sharing with Third Parties. In *2023 IEEE Secure Development Conference (SecDev)*. IEEE, 184–195.

[112] Mert D Pesé, Karsten Schmidt, and Harald Zweck. 2017. *Hardware/software co-design of an automotive embedded firewall.* Technical Report. SAE Technical Paper.

[113] Mert D Pesé and Kang G Shin. 2019. Survey of automotive privacy regulations and privacy-related attacks. (2019).

[114] Christian Plappert, Florian Fenzl, Roland Rieke, Ilaria Matteucci, Gianpiero Costantino, and Marco De Vincenzi. 2022. SECPAT: Security Patterns for Resilient Automotive E / E Architectures. In *2022 30th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. 255–264. doi:10.1109/PDP55904.2022.00047

[115] Christian Plappert, Daniel Zelle, Christoph Krauß, Benjamin Lange, Sebastian Mauthofer, Jonas Walter, Bettina Abendroth, Rasmus Robrahn, Thilo von Pape, and Hendrik Decke. 2023. A privacy-aware data access system for automotive applications. In *Proceedings of the 15th ESCAR Europe*. Berlin, Germany.

[116] Sebastian Poeplau, Yanick Fratantonio, Antonio Bianchi, Christopher Kruegel, and Giovanni Vigna. 2014. Execute this! Analyzing unsafe and malicious dynamic code loading in android applications. In *NDSS*, Vol. 14. 23–26.

[117] Vassilis Prevelakis and Mohammad Hamad. 2015. A policy-based communications architecture for vehicles. In *2015 International Conference on Information Systems Security and Privacy (ICISSP)*. IEEE, 155–162.

[118] Progressive. 2024. What is Snapshot and How You Can Save. https://www.progressive.com/auto/discounts/snapshot/.

[119] Shatad Purohit, Ayesha Madni, Arun Adiththan, and Azad M. Madni. 2023. Digital Twin Integration for Software Defined Vehicles: Decoupling Hardware and Software in Automotive System Development. In *2023 IEEE International*

Conference on Systems, Man, and Cybernetics (SMC). 1259–1264. doi:10.1109/SMC53992.2023.10394507

[120] Sampath Rajapaksha, Harsha Kalutarage, M. Omar Al-Kadri, Andrei Petrovski, Garikayi Madzudzo, and Madeline Cheah. 2023. AI-Based Intrusion Detection Systems for In-Vehicle Networks: A Survey. *ACM Comput. Surv.* 55, 11, Article 237 (Feb. 2023), 40 pages. doi:10.1145/3570954

[121] Research Gate. 2024. *Research Gate*. https://www.researchgate.net/ Last accessed March 26, 2026.

[122] Caleb Riggs, Carl-Edwin Rigaud, Robert Beard, Tanner Douglas, and Karim Elish. 2018. A Survey on Connected Vehicles Vulnerabilities and Countermeasures. *Journal of Traffic and Logistics Engineering* (01 2018), 11–16. doi:10.18178/jtle.6.1.11-16

[123] Ali Rizvi, Ani Kelkar, Philipp Kampshoff, and Sarthak Vaish. 2025. *Software-defined hardware in the age of AI.* McKinsey & Company. https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/our-insights/software-defined-hardware-in-the-age-of-ai Last accessed March 26, 2026.

[124] Marcel Rumez, Daniel Grimm, Reiner Kriesten, and Eric Sax. 2020. An overview of automotive service-oriented architectures and implications for security countermeasures. *IEEE access* 8 (2020), 221852–221870.

[125] Lu S., Ammar N., Ganlath A., Wang H., and Shi W. 2022. A Comparison of End-to-End Architectures for Connected Vehicles. In *Proceedings - 2022 5th International Conference on Connected and Autonomous Driving, MetroCAD 2022.* SOAFEE, 72–80. doi:10.1109/MetroCAD56305.2022.00015

[126] Etienne Sapin, Suraj Menon, Jingquan Ge, Sheikh Mahbub Habib, Maurice Heymann, Yuekang Li, Rene Palige, Gabriel Byman, and Yang Liu. 2023. Monitoring Automotive Software Security Health through Trustworthiness Score. In *Proceedings of the 7th ACM Computer Science in Cars Symposium* (Darmstadt, Germany) *(CSCS '23)*. Association for Computing Machinery, New York, NY, USA, Article 1, 9 pages. doi:10.1145/3631204.3631859

[127] Ankur Sarker, Haiying Shen, Chenxi Qiu, Hua Uehara, and Kevin Zhang. 2023. Brake-Signal-Based Driver's Location Tracking in Usage-Based Auto Insurance Programs. *IEEE Internet of Things Journal* 10, 12 (2023), 10172–10189.

[128] Satori. 2022. Data Anonymization: Use Cases and 6 common techniques. https://satoricyber.com/data-masking/data-anonymization-use-cases-and-6-common-techniques/ Last accessed March 26, 2026.

[129] Science Direct. 2024. *Science Direct Library.* https://www.sciencedirect.com/ Last accessed March 26, 2026.

[130] Upstream Security. 2024. Global Automotive Cybersecurity Report. https://upstream.auto/reports/global-automotive-cybersecurity-report/. Last accessed March 26, 2026.

[131] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. 2018. GIDS: GAN based Intrusion Detection System for In-Vehicle Network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. 1–6. doi:10.1109/PST.2018.8514157

[132] Khaoula Sghaier, Badis Hammi, Ghada Gharbi, Pierre Merdrignac, Pierre Parrend, and Didier Verna. 2025. Advancing Security in Software-Defined Vehicles: A Comprehensive Survey and Taxonomy. arXiv:2510.09675 [cs.CR] https://arxiv.org/abs/2510.09675

[133] Girish Shirasat and Stefano Marzani. 2024. Accelerating software-defined vehicles through cloud-to-vehicle edge environmental parity. In *aws for automotive*. SOAFEE. doi:developer/Files/pdf/white-paper/arm-aws-edge-environmental-parity-wp.pdf Last accessed March 26, 2026.

[134] Dirk Slama, Achim Nonnenmacher, and Thomas Irawan. 2023. *The Software-Defined Vehicle: A Digital-First Approach to Creating Next-Generation Experiences.* O'Reilly. https://www.digital.auto/_files/ugd/87c677_1851f6f1af6a4752a3c038038e70f222.pdf Last accessed March 26, 2026.

[135] Suzanne Smalley. 2024. *Insurer unveils policy covering drivers from connected car hacks and data leaks.* https://therecord.media/insurer-introduces-policy-covering-drivers-from-connected-car-hacks, Last accessed March 26, 2026.

[136] Fedor Smirnov, Felix Reimann, Jürgen Teich, and Michael Glaß. 2018. Automatic Optimization of the VLAN Partitioning in Automotive Communication Networks. *ACM Trans. Des. Autom. Electron. Syst.* 24, 1, Article 9 (Dec. 2018), 23 pages. doi:10.1145/3278120

[137] Ian Smith. 2024. Insurance groups urge state support for 'uninsurable' cyber risks. *Financial Times* (sep 2024). https://www.ft.com/content/c2769c6d-8bec-4167-af5c-53c6cf139851, Last accessed March 26, 2026.

[138] Jama Software. 2023. *Software Defined Vehicles: Revolutionizing the Future of Transportation.* White Paper. Jama. https://www.jamasoftware.com/whitepaper/software-defined-vehicles-revolutionizing-the-future-of-transportation-whitepaper Last accessed March 26, 2026.

[139] Springer Link. 2024. *Springer Link Library.* https://link.springer.com/ Last accessed March 26, 2026.

[140] Marco Steger, Ali Dorri, Salil S. Kanhere, Kay Römer, Raja Jurdak, and Michael Karner. 2018. Secure Wireless Automotive Software Updates Using Blockchains: A Proof of Concept. In *Advanced Microsystems for Automotive Applications 2017*, Carolin Zachäus, Beate Müller, and Gereon Meyer (Eds.). Springer International Publishing, Cham, 137–149.

[141] Latanya Sweeney. 2002. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10, 5 (Oct. 2002), 557–570. doi:10.1142/S0218488502001648

[142] Kuppusamy T. Karthik and D. McCoy. 2016. Uptane: Securing Software Updates for Automobiles. Presented at the ESCARE Europe, 2016.

[143] Pedro Veloso Teixeira, Duarte Raposo, Rui Lopes, and Susana Sargento. 2024. Software Defined Vehicles for Development of Deterministic Services. arXiv:2407.17287 [cs.DC] https://arxiv.org/abs/2407.17287

[144] Tencent Keen Security Lab. 2018. *New Vehicle Security Research by KeenLab: Experimental Security Assessment of BMW Cars.* https://keenlab.tencent.com/en/2018/05/22/New-CarHacking-Research-by-KeenLab-Experimental-Security-Assessment-of-BMW-Cars/, Last accessed March 26, 2026.

[145] The Guardian. 2023. *Nine injured in Tel Aviv ramming and stabbing attack.* https://www.theguardian.com/world/2023/jul/04/seven-injured-in-tel-aviv-ramming-and-stabbing-attack Last accessed March 26, 2026.

[146] Santiago Torres-Arias, Hammad Afzali, Trishank Karthik Kuppusamy, Reza Curtmola, and Justin Cappos. 2019. in-toto: Providing farm-to-table guarantees for bits and bytes. In *28th USENIX Security Symposium (USENIX Security 19).* Santa Clara, CA, USA. https://www.usenix.org/conference/usenixsecurity19/presentation/torres-arias

[147] Towards Automotive. 2024. Software Defined Vehicles Market Size, Share, Analysis. https://www.towardsautomotive.com/insights/software-defined-vehicles-market-sizing. Last accessed March 26, 2026.

[148] UNECE. 2021. *Uniform provisions concerning the approval of vehicles with regards to software update and software updates management system.* Regulation Addendum 155 – UN Regulation No. 156. United Nations Economic Commission for Europe, Geneva, CH. https://unece.org/sites/default/files/2021-03/R156e.pdf

[149] UNECE. 2021. *Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system.* Regulation Addendum 154 – UN Regulation No. 155. United Nations Economic Commission for Europe, Geneva, CH. https://unece.org/sites/default/files/2021-03/R155e.pdf

[150] U.S. Department of Transportation. 2020. How Connected Vehicles Work. https://www.transportation.gov/research-and-technology/how-connected-vehicles-work. Last accessed March 26, 2026.

[151] Sherin Kalli Valappil, Lars Vogel, Mohammad Hamad, and Sebastian Steinhorst. 2024. Advanced IDPS Architecture for Connected and Autonomous Vehicles. In *2024 IEEE Intelligent Vehicles Symposium (IV).* 1779–1785. doi:10.1109/IV55156.2024.10588659

[152] Volvo. 2024. Discover Volvo with Google Assistant, Google Maps and Google Play built in | Volvo Cars. https://www.volvocars.com/intl/v/connectivity/infotainment-page Last accessed March 26, 2026.

[153] Syed Wali, Yasir Ali Farrukh, and Irfan Khan. 2025. Explainable AI and Random Forest based reliable intrusion detection system. *Computers and Security* 157 (2025), 104542. doi:10.1016/j.cose.2025.104542

[154] Marian Waltereit, Maximilian Uphoff, and Torben Weis. 2019. Route derivation using distances and turn directions. In *Proceedings of the ACM Workshop on Automotive Cybersecurity.* 35–40.

[155] Bo Wang, Smruti Panigrahi, Mayur Narsude, and Amit Mohanty. 2017. *Driver identification using vehicle telematics data.* Technical Report. SAE Technical Paper.

[156] Philipp Weiss and Sebastian Steinhorst. 2023. Predictable timing behavior of gracefully degrading automotive systems. *Design Automation for Embedded Systems* 27, 1 (2023), 103–138.

[157] Bao-Fu Wu, Ren Zhong, Yuxin Wang, Jian Wan, Ji-Lin Zhang, and Weisong Shi. 2024. Vpi: Vehicle programming interface for vehicle computing. *Journal of Computer Science and Technology* 39, 1 (2024), 22–44.

[158] Oded Yarkoni. 2024. What Does the GDPR Have to Do with Car OEMs? https://upstream.auto/blog/gdpr/, Last accessed March 26, 2026.

[159] Sadia Yeasmin and Anwar Haque. 2021. A Multi-Factor Authenticated Blockchain-Based OTA Update Framework for Connected Autonomous Vehicles. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall).* 1–6. doi:10.1109/VTC2021-Fall52928.2021.9625372

[160] Longhai Yu, Xudong Bian, Yuqiao Ning, and Zhen Guo. 2023. *Software Security Analysis for Automotive Intelligent Cockpits.* IOS Press. doi:10.3233/ATDE230577

[161] Jingyuan Zhao, Wenyi Zhao, Bo Deng, Zhenghong Wang, Feng Zhang, Wenxiang Zheng, Wanke Cao, Jinrui Nan, Yubo Lian, and Andrew F. Burke. 2024. Autonomous driving system: A comprehensive survey. *Expert Systems with Applications* 242 (2024), 122836. doi:10.1016/j.eswa.2023.122836

[162] Rui Zhao, Guihe Qin, Ying Lyu, and Jie Yan. 2019. Security-Aware Scheduling for TTEthernet-Based Real-Time Automotive Systems. *IEEE Access* 7 (2019), 85971–85984. doi:10.1109/ACCESS.2019.2926113

[163] Lu Zhou, Qingrong Chen, Zutian Luo, Haojin Zhu, and Cailian Chen. 2017. Speed-based location tracking in usage-based automotive insurance. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS).* IEEE, 2252–2257.

[164] Lu Zhou, Suguo Du, Haojin Zhu, Cailian Chen, Kaoru Ota, and Mianxiong Dong. 2018. Location privacy in usage-based automotive insurance: Attacks and countermeasures. *IEEE Transactions on Information Forensics and Security* 14, 1 (2018), 196–211.