SoK: Kicking CAN Down the Road Systematizing CAN Security Knowledge

Khaled Serag
Qatar Computing Research Institute
kseragalsharif@hbku.edu.qa

Zhaozhou Tang
Georgia Institute of Technology
ztang306@gatech.edu

Sungwoo Kim
Purdue University
kim3583@purdue.edu

Vireshwar Kumar
IIT Delhi
viresh@cse.iitd.ac.in

Dave (Jing) Tian
Purdue University
daveti@purdue.edu

Saman Zonouz
Georgia Institute of Technology
szonouz6@gatech.edu

Raheem Beyah
Georgia Institute of Technology
rbeyah@coe.gatech.edu

Z. Berkay Celik
Purdue University
zcelik@purdue.edu

Dongyan Xu
Purdue University
dxu@purdue.edu

***Abstract*—For decades, the Controller Area Network (CAN) has served as the primary in-vehicle bus (IVB), extending its use to many non-vehicular systems. In recent years, CAN security has been intensively scrutinized, yielding extensive research literature. Despite its wealth, the literature lacks structured systematization, complicating efforts to assess and compare attack severity, defense efficacy, security gaps, and root causes. This leaves many defenders uncertain about the relevance of specific attacks or defenses to their systems, and even whether CAN's security problems are truly CAN-specific. As newer IVBs emerge, this matters beyond CAN: if CAN's root causes are not CAN-specific, replacing CAN may only move its problems to a new standard.**

In this paper, we systematize CAN security knowledge, presenting a comprehensive taxonomy and assessment models of attackers, attacks, and defenses. We identify replicable attacks and defense gaps, and investigate their root causes to determine their exclusivity to the CAN standard. We then investigate whether those root causes appear in three emerging IVBs and assess their effectiveness in solving fundamental CAN security problems. Our findings challenge common perceptions: CAN is more securable than perceived, most of its insecurity root causes are shared across IVBs, and merely adopting newer IVB technology does not solve persistent security problems. We conclude by suggesting that securing future in-vehicle communication requires addressing shared root causes, and we propose four research directions with the most promising potential.

I. Introduction

In the 1970s, integrated circuits began appearing in vehicles as Electronic Control Units (ECUs) to digitize certain functions. By the 1980s, vehicles contained numerous ECUs, but point-to-point connections introduced significant wiring complexity. In 1986, Bosch released the Controller Area Network (CAN) standard, tackling issues including wiring, interference resistance, and decentralization, enabling modern vehicles to feature dozens of ECUs, coordinating essential functions. Today, CAN is present in all vehicles and has extended its use to industrial automation, avionics, medical devices, and many other arenas.

Network and Distributed System Security (NDSS) Symposium 2027
22–26 March 2027, Seoul, Republic of Korea
ISBN 978-1-970672-09-1
https://dx.doi.org/10.14722/ndss.2027.230993
www.ndss-symposium.org

Designed in an era of network isolation, increasing connectivity has exposed CAN to many exploits, triggering a research race between attack and defense, and resulting in very rich and diverse literature. While rich in technical detail, it predominantly emphasizes technical novelty and how each paper differs from closely related ones—often making papers pursuing the same end goal look unrelated. As a result, the broader picture remains diffuse, especially from a defense standpoint. A defender starts from a priority-ranked hierarchy of security objectives (e.g., availability of a safety function, integrity of a particular sensor value, or confidentiality of certain data) their system must preserve, assuming a particular threat model. They need a way to turn the vast literature into something usable: which attack papers target the same security objective, what is replicable on their systems, what threat models they presuppose, what performance impact a defense entails, and what defense gaps they should know about.

In the absence of a systematization framework that could help with these questions, the wealth of CAN security literature has paradoxically worked to its disadvantage. This may cultivate a perception of CAN as hopelessly insecurable, especially as some voices explicitly call for replacing CAN with newer in-vehicle bus (IVB) technologies, including CAN evolutions and alternative buses, with security among the primary stated motivations [1], [2]. Several valuable surveys provided nascent systematization attempts [3], [4], [5], some focusing on classifying attackers based on attack surfaces and entry points [6], [7], [8], and some on specific defense approaches [9], [10] or authentication protocols [11]. Nonetheless, a coherent and comprehensive systematization framework remains conspicuously absent. Without a framework to articulate attacker privileges, map privileges to attacks, group attacks with similar security objectives, and assess the significance of an attack or a defense, it is infeasible to identify replicable attacks, defense gaps, or root causes. Finally, no prior research systematically analyzed whether CAN security problems are exclusive to the CAN standard, or whether they stem from broader structural limitations of in-vehicle communication. As interest in new IVB technologies grows, the issue becomes more pressing. Misunderstandings about CAN's core security issues may foster

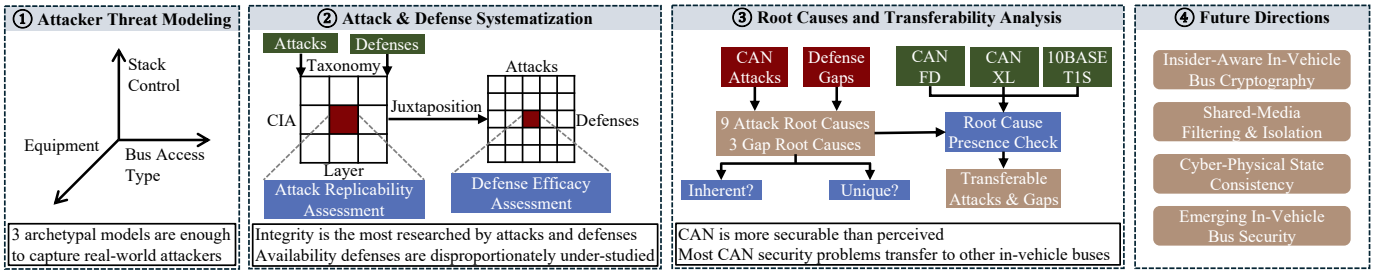

Fig. 1. Overview of the systematization flow and the paper's main contributions.

the belief that simply adopting newer technology resolves CAN security issues, unknowingly reproducing them in other IVBs.

This paper systematizes existing CAN security knowledge, presenting a comprehensive defender-oriented systematization framework. We take this perspective because OEMs, practitioners, researchers, and standards bodies are, to varying extents, all defenders seeking to make informed decisions. Our systematization serves each directly. OEMs can anticipate which security problems transfer when adopting a new technology. Practitioners can determine which attacks are possible in their systems and which defenses address them. Researchers can focus their efforts on gaps instead of already crowded areas. Standards bodies can identify the root causes underlying security weaknesses in standards.

As shown in Fig. 1, we start with multi-dimensional attacker modeling, contextualizing ability, access, and equipment, to understand the various attacker models appearing in the literature. Second, we provide an objective-based taxonomy and assessment model for the attacks. The goal is to understand the various attack objectives appearing in the literature, the different variants, and their replicability across CAN systems in terms of their difficulty and required attacker models or implementation flaws. We do the same for defenses, taxonomizing the existing defense landscape and assessing their effectiveness in tackling the taxonomized attacks in terms of security, infrastructure, and performance impact. Juxtaposing attacks and defenses aims to expose what is most targeted, what is most defended, which attacks have defense gaps, and where research is concentrated or lacking. Third, building on these outcomes, we then analyze the root causes behind CAN attacks and defense gaps to understand the extent to which they are inherent or unique to CAN. We then conduct an investigation of three representative emerging IVBs: two CAN evolutions, CAN FD and CAN XL, and one alternative bus, 10BASE-T1S. The goal is to determine the extent to which CAN problems are transferable to other IVBs and the extent to which these IVBs solve CAN security issues. Finally, we identify four research directions with the highest potential to improve the security of in-vehicle communication.

Our study results in interesting outcomes. Relative to threat models, our multi-dimensional analysis reveals that the seemingly diverse attacker models used across the literature could collapse into only three attacker models that are enough to adequately represent the various models in real life. This reduction is small enough to be actionable while preserving the distinctions that matter for security. It allows attacks to be compared under common attacker models and gives practitioners a direct mapping from attacker privileges to feasible attacks and applicable defenses. It also exposes cases where a defense appears effective only because it assumes a weaker attacker. In terms of research interest, while integrity receives the most attention in the attack literature, it also receives the highest attention in the defense literature, yielding several effective defenses. On the other hand, availability appears to be the most exposed CIA element, with the highest proportion of highly replicable attacks with no effective defenses and the least amount of research papers. In terms of root-cause and transferability analysis, we reveal that CAN is more securable than perceived, that most CAN insecurity problems are rooted in causes inherent to all single-channel communication media or vehicular systems, that most of them are transferable to other IVBs, and that none of its defense gaps are rooted in causes unique to CAN. Finally, we suggest that securing CAN evolutions, and all future in-vehicle communication down the road, requires addressing shared root causes more than CAN-specific causes. Overall, we make the following contributions:

- We develop a defender-oriented CAN security systematization framework, including a CAN-oriented abstraction stack, a multi-dimensional attacker model, and assessment criteria for attacks and defenses.
- We survey, taxonomize, and assess CAN attacks and defenses, covering attack goals, attack replicability, attack variants, defense approaches, and defense effectiveness.
- We juxtapose attacks and defenses to identify highly replicable attacks, expose defense gaps, and show where research is concentrated or lacking.
- We analyze the root causes behind CAN attacks and defense gaps to determine the extent to which they are inherent or unique to the CAN standard.
- We investigate whether CAN 2.0 security problems transfer to other IVBs through a targeted investigation of three representative emerging IVBs: CAN FD, CAN XL, and 10BASE-T1S.
- We identify four research directions with the highest potential to improve in-vehicle communication security.

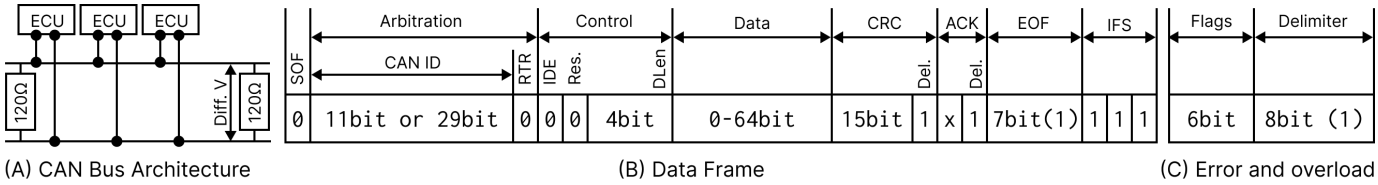

Fig. 2. CAN bus architecture, data frame, and error & overload frame formats.

II. CAN Basics

CAN Operation. Fig. 2-A illustrates the masterless, publish-subscribe nature of the CAN bus, consisting of two wires terminated by resistors. Their voltage difference represents bits: zeros (dominant) and ones (recessive). Nodes may initiate transmission when the bus is idle, transmitting data bit-by-bit. Fig. 2-B illustrates the format of a data frame. If two nodes transmit simultaneously, collision is resolved using arbitration over the ID and RTR fields: when one node sends a 0 and another a 1, the node transmitting 1 halts, yielding to the 0 transmitter. Lower IDs thus have higher priority. While nodes may transmit multiple message IDs, each ID is typically associated with only one ECU. CAN frames begin with a Start of Frame (SoF) bit (logical 0) and conclude with 7 End-of-Frame (EoF) bits (logical 1). A minimum Interframe Space (IFS) of 3 bits separates consecutive frames on the bus.

Error and Overload Handling. CAN defines 5 error types. Nodes update a transmit (TEC) and a receive error counter (REC). Transmit errors increase the TEC by 8. Others increase REC. If either counter exceeds 127, nodes enter the error-passive state with stricter transmission rules. If transmit errors continue (TEC $>$ 255), they move into the bus-off state and stop participating in communication. Nodes signal errors using an error frame composed of a flag and a delimiter (Fig. 2-C). In the error active state, the flag is dominant, interrupting ongoing transmission. In the error passive state, it is recessive, invisible, and cannot interrupt transmission. Overload frames have a similar format to active errors, but do not increase error counters and occur only between frames by resource-constrained nodes to slow down bus communication.

CAN Stack Model. CAN aligns with the two bottom OSI layers. To communicate, a typical ECU (Fig. 3) uses a CAN controller (enforces protocol rules) connected to a transceiver (converts 0s and 1s to differential voltage). Although the functionalities of the controller and transceiver highly align with OSI's data-link and physical layers, the alignment is not exact. Consequently, we choose to model CAN stack after the typical components of an ECU to better fit CAN's nature:

- *Transwire Layer:* Encompassing the transceiver and wires, it roughly corresponds with the OSI's physical layer.
- *Controller Layer:* Encompassing only the CAN controller, this layer roughly corresponds with the OSI's data-link layer.
- *Higher Layer:* Higher-layer protocols that extend CAN functionality to higher OSI layers (e.g., transport, session, etc.) are not part of the CAN standard, but are part of the CAN stack. We consider them collectively as the "higher layer", encompassing all that interacts with the controller from above.

Common Higher-Layer Protocols. Many protocols can run on top of CAN. Two common examples are UDS and J1939. *1) Unified Diagnostic Services (UDS):* a client-server diagnostic protocol in which a client, typically a diagnostic tool connected through the OBD-II port, communicates with ECUs acting as servers. UDS supports privileged operations such as reading or modifying ECU memory and firmware, and disabling ECU functionalities. These operations can be protected using seed-key authentication. *2) J1939:* a protocol widely used for communication and diagnostics in trucks. It supports broadcast and request-response communication, packet fragmentation across multiple CAN frames for payloads exceeding 8 bytes, and dynamic higher-layer source-address assignment through an address-claim procedure after network initialization.

III. Methodology & Scope

Taxonomy. To group attack papers that appear disparate in the literature by their underlying attack objective, we first place the paper in a coarse target bin defined by the intersection of the CIA triad and CAN stack layer (Sec. II) for each attack a paper proposes. A paper may thus appear in multiple bins if it proposes attacks against different targets. Within each bin, we identify the specific attack objectives and group papers pursuing the same objective into one attack category, even when their execution paths differ. This groups variants of the same attack goal and separates attacks rooted in lower-layer standard properties from those exploiting higher-layer protocols or implementations, which later supports root-cause analysis.

Our defense taxonomy uses the same CIA $\times$ layer grid to enable direct juxtaposition with attacks. Since a defense may cover several properties, we taxonomize each defense idea by its primary protected CIA property and the deepest protected layer within that property, while noting protection against other objectives when relevant. A paper proposing multiple defense ideas may appear in multiple bins. This exposes misleading coverage, such as higher-layer integrity protection masking a controller-layer integrity weakness, and reveals where defenses are crowded, missing, or ineffective.

Inclusion & Exclusion Criteria. Our systematization covers security knowledge on CAN 2.0. We include attacks launched by devices directly interfacing with the bus and targeting the confidentiality, integrity, or availability of the bus, its nodes, or the system it operates within (e.g., a vehicle). We include defenses developed for these attacks, and some CAN-adaptable solutions proposed for non-CAN systems. We exclude attacks

launched by non-bus-interfacing devices, such as attacks on external vehicle sensors by a non-bus-connected device.

Paper Selection Methodology. We followed a two-stage approach consisting of an initial database search followed by manual citation-graph traversal. We searched for papers published between 1983, when Bosch began developing CAN, and June 1, 2025 in eight major security conferences: NDSS, IEEE S&P, USENIX, CCS, ACSAC, ASIACCS, RAID, and EuroS&P, using the keywords "vehicle" and "Controller Area Network." This yielded 123 papers, of which 28 met our inclusion criteria and formed the initial seed. We then traversed their cited and citing papers without restricting venue or type, including relevant industry white papers and technical reports.

As the citation traversal expanded beyond the initial venues and across four decades of research, the candidate literature grew substantially, encompassing numerous works from specialized and smaller venues, many of which repurposed established attack and defense ideas with minor variations. We therefore retained only papers introducing a distinct attack or defense according to our taxonomy. Specifically, papers with the same bin, purpose, and assessment were treated as variations of the same idea, in which case we retained the earliest publication. We repeated the traversal until no additional eligible papers were identified, resulting in 150 papers.

Paper Category Breakdown. Of the 150 papers, 43 proposed attacks only, 100 defenses only, and 7 both. Of the 100 defense-only papers, 17 were dual-use, meaning their techniques could also be used by attackers and were therefore additionally included in the attack taxonomy. In total, 67 papers contributed to attacks and 107 to defenses. Among attack papers, 33 targeted confidentiality, 28 integrity, and 23 availability, while 29 targeted the higher layer, 34 the controller layer, and 21 the transwire layer. Excluding dual-use papers, these counts were 17, 28, and 22 by security element and 29, 32, and 6 by layer, respectively. Among defense papers, 19 mainly addressed confidentiality, 88 integrity, and 5 availability, while 39 provided protection down to the higher layer, 70 the controller layer, and 17 the transwire layer. Note that counts do not sum to the corresponding paper totals because a paper may propose multiple distinct attacks or defenses that fall into different bins.

Outside of the taxonomized papers, we additionally referenced relevant sources that do not propose novel CAN attacks or defenses. This included standard specification documents, review papers, news articles, user manuals and presentations related to the CAN ecosystem. We have provided a spreadsheet detailing each paper and its categorization at [12].

IV. Attacker Threat Modeling

Several surveys classify attackers based on bus access type only [3], [6], [7], [8]. For a deeper understanding, we model attackers along three dimensions: *bus access type* (Sec. IV-A.1), *equipment* (Sec. IV-A.2), and *control* over CAN stack layers (Sec. IV-A.3). A key revelation of this modeling is that, rather than producing a large number of attacker types, CAN attackers can be effectively reduced to only three archetypal models, discussed in Sec. IV-B, that we use throughout the paper.

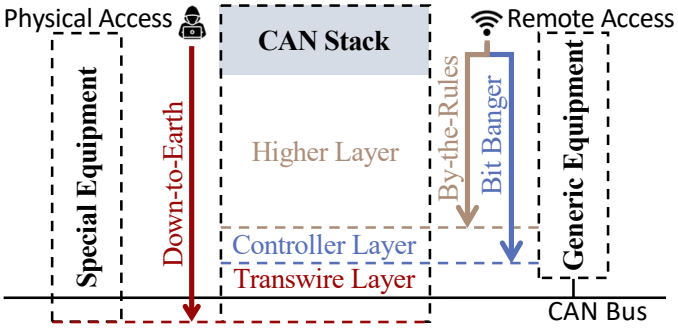

Fig. 3. Typical ECU components, CAN stack, and archetypal attacker models.

A. Modeling Dimensions

1. Bus Access Type ($\mathcal{A}_c$). Two levels appear in the literature:

■ *Remote/On-ECU (○):* The weakest access type as the attacker has no physical access. The attacker here cannot attach any equipment and has to utilize the capabilities of an existing ECU remotely. Researchers showed this is achievable by compromising communication between smartphone apps and vehicles [13], [14] or wireless OBD-II dongles [15]. Further, the growing connectivity of ECUs exposes new remote attack surfaces, such as Wi-Fi, Cellular, Bluetooth, remote keyless entry, TPMS, and radio [16], [17], [18], [19], [20].

■ *Physical/Side-ECU (●):* The attacker has physical access to the bus that could be brief, sufficient to install a malicious device or a side ECU that may have wireless capabilities. Examples include accessing the OBD-II port during routine maintenance or service, or breaking through exposed areas such as behind the bumper or headlights [21]. More drastic changes, such as disconnecting ECUs or altering the bus topology require extensive access, which is difficult as it is often limited to authorized users.

2. Equipment ($\mathcal{E}_q$). Two broad categories are used in attacks:

■ *Generic (○):* The standard off-the-shelf components found in mainstream ECUs. Any access type could use it but it is the only equipment type remote attackers can use as installing other equipment requires physical access. However, attackers may alter ECUs' logical connections through software, to deviate from the intended model (Fig. 3 and Sec. IV-A.3).

■ *Special (●):* The attacker designs their own equipment with its hardware layout fitted specifically for their intended attack. Physical access is required to install this equipment.

3. Stack Control. The deepest layer of the CAN stack (Fig. 3) the attacker could penetrate and control:

■ *Higher Layer:* The attacker only controls the higher layer and has no control over the protocol controller or transceiver, except for what the higher-layer allows (e.g., baudrate). This control level is achievable using the weakest access type and equipment: remote access and generic equipment.

■ *Controller Layer:* Attacker has full or partial control down to the controller layer, directly managing bus interactions. Traditionally, this required special hardware and physical access [22], [23], [24]. However, research [25], [26], [27]

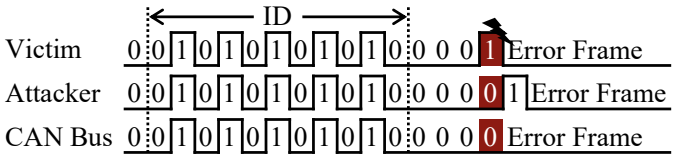

Fig. 4. Inducing an error using the simultaneous transmission technique.

shows that it is remotely attainable on many generic ECUs by exploiting peripheral clock gating to manipulate CAN frames mid-transmission, special controller modes, or pin conflicts to connect other peripherals, such as the SPI, I2C, or UART, directly to the CAN transceiver pins.

■ *Transwire Layer:* Using special equipment and physical access, the attacker controls the entire CAN stack.

B. Archetypal Attacker Models

The 3 outlined modeling dimensions contain 2, 2, and 3 possibilities, respectively. If simply multiplied, they would produce 12 combinations. However, most combinations are not meaningfully distinct because these dimensions are interdependent. The defining dimension is stack control: controlling a given CAN stack layer unlocks a specific set of attack tool-blocks, which form the building blocks used to launch the attacks in the literature. How control is achieved, through equipment or bus access type, affects feasibility rather than the resulting tool-block set. Moreover, control is cumulative: an attacker that controls a deeper layer inherits the tool-blocks of all higher layers. Thus, many apparent attacker types collapse into three equivalence classes defined by the deepest layer controlled. Under our stack model, this yields three archetypal attacker models (Fig. 3), listed below.

1. The By-the-Rules Attacker (○). As shown in Fig. 3, this model is defined by its higher-layer control. It requires at least remote access and generic equipment. Due to its limited stack control, the attacker abides by all transwire and controller layer rules, confining the tool-blocks at their disposal to:

■ *Frame Sniffing:* Reading messages assembled and processed by their uncompromised controller, without bit visibility.

■ *Frame Injection:* Injecting full data and remote request, but not error, overload, or malformed frames, only when the bus is idle through their intact controller and transceiver with full control over the data, ID, and DLC fields.

■ *Simultaneous Transmission:* This tool-block allows a node to bypass the collision resolution (arbitration) mechanism of CAN and transmit simultaneously with another node to achieve various purposes. It was discovered by Cho et al. [28]. As shown in Fig. 4, if two nodes transmit two frames exactly at the same time with the same ID, both will transmit simultaneously until the first different bit, past the arbitration field. There, simultaneous transmission ends, resulting in an error frame (active or passive), and the termination of at least one transmission.

2. The Bit-Banger (◐). As shown in Fig. 3, this model is defined by its control over the controller layer, requiring a minimum of remote access and generic equipment. The attacker here directly manages bus interactions typically handled by the CAN controller including encoding and rule enforcement. Attackers still adhere to transwire layer constraints (e.g., cannot change or read voltage levels, rise and fall times, etc.) as they do not control it. In addition to all the tool-blocks of by-the-rules attackers, it has this additional set:

■ *Bit and Pulse Injection:* Attacker may inject individual bits or pulses. This may happen while the bus is idle or during another transmission, including performing 1→0 bit-flips.

■ *Dynamic Bit-width and Baudrate Manipulation:* The attacker can dynamically control the width and baud of bits. While by-the-rules attackers could set their CAN controllers' baudrates to various rates, they cannot do it mid-transmission.

■ *Bit and Pulse Sniffing:* Sniffing bits and pulses directly from the bus, including random pulses or individual bits transmitted during idle time, or within valid or invalid frames.

3. The Down-to-Earth Attacker (●). As shown in Fig. 3, the attacker has full stack control down to the transwire layer, achievable only through physical access to attach special equipment or a side-ECU. In addition to the tool-blocks of the two previous models, it could manipulate all CAN rules (e.g., inject non-standard voltages, read exact voltage levels, rise and fall times, etc.).

V. Attacks On CAN

In this section, we taxonomize CAN attack literature as described in Sec. III. We assess how likely these attacks are to recur across CAN-based systems by assessing their replicability beyond the original study setting using the criteria listed below. Tbl. I summarizes the resulting taxonomy, showing how attack targets distribute across CIA and layer, along with each attack's replicability assessment. In the table, highly replicable attacks are highlighted in light brown and blue for by-the-rules attackers and bit-bangers, respectively. We provide a spreadsheet at [12] detailing the rationale for each assessment.

Replicability Assessment Criteria. Our goal is to help defense analysts judge whether a reported attack is likely to affect their CAN-based systems, enable uniform cross-paper comparison, and support later root-cause analysis. Therefore, we use replicability to assess CAN attacks: a system-agnostic measure of how readily an attack can be reproduced beyond its original study setting. We adopt this criterion instead of impact because impact depends heavily on the target function and deployment context. Below, we detail our replicability criteria and metrics:

■ *Vulnerability Restrictiveness ($\mathcal{V}_r$):* The vulnerability underlying the attack could be *standard (○)*: a part of CAN specifications and every system has it if there are no security measures; *widespread (◐)*: not part of the standard but present on many CAN systems; *restricted (●)*: local only to a certain system, ECU, or higher-layer protocol implementation.

■ *Difficulty ($\mathcal{D}_f$):* An exploit is *easy (○)* if it does not require pre-existing knowledge or continuous high-overhead computation. It is *moderate (◐)* if it requires pre-existing knowledge, or continuous high-overhead computation. It is *difficult (●)* if it requires compromising more than one ECU, obtaining extensive physical access, or builds upon another restricted vulnerability or difficult attack.

TABLE I
CAN ATTACK TAXONOMY AND ASSESSMENT
HL: HIGHER LAYER, **CT**: CONTROLLER LAYER, **TW**: TRANSWIRE LAYER

CIA	**Confidentiality**					**Integrity**									**Availability**						
Layer	**HL**		**CT**	**TW**		**HL**			**CT**					**TW**	**HL**		**CT**				**TW**
Attack	C1. Communication Surveillance	C2. Memory Leakage	C3. Network Reconnaissance	C4. HW/Topology Profiling	C5. EM Emissions	I1. Fake Data Fabrication	I2. Mem./FW Manipulation	I3. Bypass Diag. Authentication	I4. Impersonation & Replay	I5. Frame Hijacking	I6. Double Receive	I7. Unorthodox Frames	I8. Poly-Semantic Frames	I9. Frame Tampering	I10. Physical FP Distortion	A1. ECU Func. Disabling	A2. Sybil Exhaustion	A3. Error Injection	A4. Error State Manipulation	A5. Bus Access Denial	A6. Synchronization Disruption	A7. Signal Attenuation
Attacker Model ($\mathcal{A}_m$)	○	○	○	●	●	○	○	○	○	○	◐	◐	◐	◐	○	○	○	○	○	○	◐	◐
Vuln. Restrict. ($\mathcal{V}_r$)	○	●	○	○	○	○	●	●	○	○	○	○	◐	◐	○	◐	◐	○	○	○	○	○
Difficulty ($\mathcal{D}_f$)	◐	○	○	◐	◐	◐	○	◐	○	○	○	○	◐	◐	○	○	○	○	○	○	○	●

■ *Weakest Attacker Model ($\mathcal{A}_m$):* The weakest archetypal attacker model (Sec. IV-B) required to launch the attack.

Highly Replicable Attack. We consider an attack highly replicable if it exploits a standard vulnerability ($\mathcal{V}_r$), has easy to moderate difficulty ($\mathcal{D}_f$), and is remotely launchable (i.e., its weakest attacker model ($\mathcal{A}_m$) is a bit-banger or a by-the-rules attacker).

A. Confidentiality and Privacy Attacks

Target: Higher Layer

C1. Communication Surveillance. On a broadcast bus, any attacker could surveil bus messages ($\mathcal{V}_r$: ○). However, the attacker needs format knowledge or reverse engineering to understand their meaning as most vehicle manufacturers keep formats proprietary ($\mathcal{D}_f$: ◐). While a full understanding of the proprietary formats often requires full physical access using special equipment [29], research shows that an adequate understanding only requires sniffing the entire bus messages or having a trace-dump and then observing correlations, message priorities, fuzzing, probing [30], [31], [32], [33], [34], or using third-party debuggers and phone companion apps [35], [36]. Public datasets of reverse-engineered message formats of many vehicle models are available [37]. Attackers could obtain various private system and personal information from correlating transmitted sensor data, including driver location and vehicle identification, driving patterns profiling, and trip information [15], [34], [38], [39]. In some systems that reuse the same secret keys for ECU diagnostic authentication (Sec. II), attackers could sniff and replay the keys to bypass authentication.

C2. Memory Leakage. By-the-rules attackers can exploit vulnerable higher-layer protocol implementations ($\mathcal{V}_r$: ●) to reveal ECUs' memory contents. They could issue diagnostic commands (Sec. II) to access an ECU's sensitive memory, extract keys, or even sometimes dump the entire firmware [35], [40], [41]. However, this requires them to successfully bypass the ECU's diagnostic authentication first (another attack explained in I3). Alternatively, Chatterjee et al. [42] showed that attackers could reveal memory contents of a victim sender-ECU that uses higher-layer protocols allowing packet fragmentation (e.g., J1939, Sec. II). In such multi-frame sessions, the sender first announces the intended transfer length, and the receiver responds with flow-control information that tells the sender what to transmit next. An attacker on the receiving ECU can manipulate the response message, such as requesting more frames than the announced transfer length. In vulnerable implementations, this may cause the victim to transmit additional unintended data beyond the message boundary.

Target: Controller Layer

C3. Network Reconnaissance. The source ECUs of messages are proprietary information of the manufacturer, but by-the-rules attackers could identify them using various techniques. One technique exploits loopholes in CAN's error handling mechanism ($\mathcal{V}_r$: ○) [24]. The attacker first pushes an ECU to the error-passive state using the error-state manipulation technique (A4), then observes bus messages that display an error-passive behavior. The same technique could also be used as a defense to identify the source of malicious messages [43]. Alternatively, reconnaissance can be performed passively using a technique that was initially published as a defense [44], [45]. The attacker continuously monitors and profiles the cumulative clock skews (small drift in cyclical transmission time) of periodic messages. Messages with the same transmitters will have similar clock skews. However, with generic ECU equipment, accurate and continuous monitoring usually requires high overhead. Moreover, this technique can be effectively evaded if ECUs carefully control their transmission time to mimic other nodes [46], [47].

Target: Transwire Layer

C4. Hardware and Topology Profiling. Due to hardware manufacturing differences, different ECUs' messages often exhibit distinct voltage/time characteristics. Accordingly, down-to-earthers could infer information about an ECU's hardware by profiling its voltage characteristics using an oscilloscope or analog-to-digital converter (ADC) [48], [49], [50], [51], [52], [53], or by measuring the time characteristics of rising and falling edges in a frame using a time-to-digital converter (TDC) [54], [55], [56], [57]. Moreover, physical layer features are influenced by ECU locations and bus topology. This allows attackers to identify ECU locations or bus topology changes (e.g., ECU addition or removal) using a voltage sampler [58], [59], [60], or by connecting wiring from both ends of the bus to a TDC and monitoring the difference in signal arrival time at the two ends [61], [62].

C5. Electromagnetic Emissions. Down-to-earthers can inter-

cept electromagnetic waves of the CAN bus without being physically connected. From up to 1 meter away using a probe and RF analyzer, researchers showed that CAN electromagnetic emissions are interceptable with enough clarity allowing full reconstruction of frames and opening another door to communication surveillance (C1) attacks [63].

Confidentiality Takeaway: Confidentiality is least targeted in pure attack papers (17/50), but most targeted when dual-use defense papers are included (33/67). This shift comes from attacker-identification defenses, whose fingerprinting/profiling methods could be exploited by attackers. It also has the highest share of standard-rooted attacks: 4 of 5 categories.

B. Integrity Attacks

Target: Higher Layer

I1. Fake Data Fabrication. A by-the-rules attacker controlling an ECU could falsify the data this ECU is supposed to send, including sensor signals, commands, etc. [13], [14], [15], [16], [17], [18], [19], [20]. However, to achieve a controlled and intended impact, the attacker must know the format of the data/commands that need to be fabricated. This usually requires reverse engineering ($\mathcal{D}_f$: ◐). Researchers showed that with careful fabrication, they can control vehicle functions such as the radio, engine, brakes, steering, acceleration, etc. [35], or manipulate higher-layer protocols (e.g., UDS, Sec. II) [64], [65].

I2. Memory and Firmware Manipulation. By-the-rules attackers could modify a victim ECU's memory to tamper with its data or manipulate the firmware's control flow. This can be achieved by exploiting diagnostic services that allow for memory and firmware modification (e.g., UDS RequestDownload or WriteBlock services, Sec. II). However, this is only possible if the ECU does not implement diagnostic authentication ($\mathcal{V}_r$: ●) [35], or its diagnostic authentication can be bypassed as explained in I3. An alternative approach is to cause buffer overflow in ECUs with vulnerable higher-layer protocols that implement multi-frame sessions (e.g., J1939, Sec. II). The attacker may send more packets than the negotiated number at the beginning of the session [42] or reset the number of frames expected by the receiver to a smaller number [66], so the receiver writes past the boundaries allocated for the session.

I3. Bypassing Diagnostic Authentication. Researchers investigated ways to bypass diagnostic authentication (Sec. II). For by-the-rules attackers, methods including replaying authentication messages obtained from off-the-shelf diagnostic tools [67] and brute-forcing UDS passwords [41] could work on some ECUs ($\mathcal{V}_r$: ●). For down-to-earthers, more intrusive methods such as applying a voltage glitch to the ECU's processor so it skips authentication instructions [40] were successful for some ECUs. Lauser et al. [68] proposed to bypass authentication by hijacking a session post authentication: an OBD-II adapter connecting an external legitimate device to the ECU may act initially as a man-in-the-middle attacker, then hijack the session.

Target: Controller Layer

I4. Impersonation and Replay. By-the-rules attackers could impersonate another ECU by sending or replaying messages carrying an identifier of the impersonated victim ECU [67], [13], [14], [15], [16], [17], [18], [19], [20]. Since CAN does not have an explicit source ID field, the identifier is usually the CAN ID as each CAN ID is typically only transmitted by one ECU. In higher-layer protocols with an explicit source ID field, they could forge the source ID (e.g., J1939, Sec. II) [42], [66], [69]. By-the-rules attackers may be able to perform such impersonation while successfully evading detection by some types of Intrusion Detection Systems (IDS) [44], [46], [70].

I5. Frame Hijacking. If an attacker injects a dominant bit when an error-passive victim is transmitting a recessive bit, the victim's error frames will be invisible to other ECUs. This allows the attacker to finish transmitting the frame, effectively overwriting a part of the victim's message [71]. This could achieve various purposes (e.g., if a secret authentication token is transmitted at the beginning of the message, the attacker can change the message's content while keeping the token intact). This attack can be launched by bit-bangers or by-the-rules attackers invoking simultaneous transmission (Sec. IV-B.1).

I6. Double Receive. Per the standard, CAN receivers validate a frame on receiving the second-to-last EOF bit without errors. Transmitters consider it valid if the last bit contains no errors. A bit-banger could inject an error in the last EOF bit to exploit this standard mismatch ($\mathcal{V}_r$: ○) [71]. This makes the transmitter believe the message was not received successfully although it was. In most cases, the transmitter retransmits the message. This could achieve various purposes (e.g., cause a desynchronization when nodes need to agree on the counter values in messages).

I7. Unorthodox Frames. Certain fields in a CAN frame are not completely free to set. For example, the Data Length Code (DLC) should match the actual message length. However, when they are not set correctly, the standard does not specify how they should be detected with the error handling mechanism or provide clear guidelines on how receivers should react ($\mathcal{V}_r$: ○). Bit-bangers could exploit these under-defined parts of the protocol to cause or explore undefined receiver behavior [72].

I8. Poly-Semantic Frames. CAN controllers interpret frames by reading voltage samples at periodic sample points. Controllers' sample points are not necessarily the same ($\mathcal{V}_r$: ◐). A bit-banger that knows at least two ECUs' sample points ($\mathcal{D}_f$: ◐) could fabricate frames with voltage transitions that fall between the ECUs' sample points so different receivers interpret the frames' content differently [73], [74]. However, to make all interpretations of their frames accepted by receivers, they must craft the CRC field so that it is valid for all interpretations. Similarly, CANflict [26] showed that many commercial ECUs allow disconnecting the CAN controller from the CAN transceiver and connecting it instead to other peripherals including the SPI, UART, or I2C controllers. Bit bangers can transmit messages whose physical form is valid under different communication protocols (e.g., CAN and SPI), but with different meanings. This could be used to achieve various purposes, such as exchanging secret messages that may contain a certain payload when interpreted as a CAN frame, but a different payload when interpreted as another protocol.

I9. Frame Tampering. Attackers may tamper with frames in transit to cause receivers to see a different message from the one sent by the legitimate transmitter. Due to the parasitic reactance of the bus ($\mathcal{V}_r$: ○), Mohammed et al. [75] discovered that 8 bit-bangers ($\mathcal{D}_f$: ●) inducing transient signals near a receiver may flip bit values, including 0→1, on the receiver's side only. The transmitter will not raise a bit error, as it does not detect the flip. The flips should encompass the CRC to accord with the frame, constituting a form of MITM. Tang et al. [72] proposed a method requiring only 1 ECU which relies on sample point discrepancy ($\mathcal{V}_r$: ◐, $\mathcal{D}_f$: ◐). The attacker injects pulses between the receiver and transmitter sample points to make the receiver read a different message while not triggering the transmitter's error detection. While 1→0 flips are easy to achieve as CAN physical layer permits it, Rogers et al. showed that one down-to-earther may perform 0→1 flips, allowing it to make active error frames disappear [76].

Target: Transwire Layer

I10. Physical-Fingerprint Distortion. By-the-rules attackers could change the physical characteristics of the messages they transmit over the wire to achieve various purposes. They may evade IDSs that use these physical characteristics to detect attacks (physical IDS, explained in DI7) such as impersonation (I4). For example, by carefully controlling their transmission time ($\mathcal{D}_f$: ◐) they may evade IDSs that use clock skews of periodic messages [46], [47]. Two by-the-rules-controlled ECUs ($\mathcal{D}_f$: ●) invoking simultaneous transmission (Sec. IV-B.1) [70] or a single bit-banger ($\mathcal{D}_f$: ○) [72] can distort an ECU's voltage signature as the resulting signal will be the superposition of the two transmitters' signals. The attacker first does that in the training or the online update period of the IDS to poison its training set to learn the superposition voltage as the victim's signature.

Integrity Takeaway: Most targeted, most diverse, and most replicable. Integrity is the most targeted CIA element, appearing in 28 of 50 papers, and has the highest attack diversity, with 10 categories. It also has the highest number and proportion of highly replicable attacks: 6 of 10.

C. Availability Attacks

Target: Higher Layer

A1. ECU Functionality Disabling. By-the-rules attackers may disable many ECUs directly by sending higher-layer commands (e.g., UDS, Sec. II) to the engine [77], brakes, power steering, ignition, lights [35], wipers, wireless door locking [15], and other vehicle systems [16], [19], [67], [69]. Attackers can know which diagnostic commands to use by reverse engineering off-the-shelf diagnostic tools that are made for the specific vehicle make and model. Alternatively, they could computationally overwhelm a victim ECU to make it unable to perform its main functionality (e.g., by leading it to miss real-time deadlines). Mukherjee et al. [66] showed that attackers can achieve this exploiting higher-layer protocols that support request-response dynamics (e.g., J1939, Sec. II. $\mathcal{V}_r$: ◐) by simply sending a high number of requests to a victim ($\mathcal{D}_f$: ○).

A2. Sybil Exhaustion. Protocols supporting sessions, or that allow ECUs to claim higher-layer addresses (e.g., J1939, Sec. II. $\mathcal{V}_r$: ◐) may be vulnerable to starvation attacks. Attackers can claim the same address as one or more victims to prevent them from getting the desired address [78]. Similarly, establishing connections from fake sources with a victim could prevent it from establishing new legitimate connections [42].

Higher Layer Takeaway: Heavily studied but less replicable. Higher-layer attacks appear in 29 of 50 attack papers, second only to controller-layer attacks, yet have a low replication likelihood. They rely least on standard-rooted vulnerabilities and most on restricted ones. Only 2/7 are highly replicable.

Target: Controller Layer

A3. Error-Injection. Attackers could cause errors in messages transmitted by a victim ECU by inducing controlled collisions to flip a bit from 1 to 0 anywhere past arbitration. This results in an error frame whose direct impacts are destroying the frame and increasing the victim's error counters (Sec. II, $\mathcal{V}_r$: ○). To do so, by-the-rules attackers could use the simultaneous transmission technique, explained in the by-the-rules model in Sec. IV-B.1. Bit-bangers could flip the bit directly. Error injection could be used for testing [23], to destroy frames, or as a stepping stone for other attacks (e.g., A4).

A4. Error State Manipulation. Attackers could perform repeated error injection (A3) to change a victim's error counter and push it to an error state of their choosing ($\mathcal{V}_r$: ○). They can push a victim to the error-passive state to prepare for subsequent attacks (e.g., C3, I5, and I10), or to the bus-off state to achieve persistent DoS of the victim. Error state manipulation was first proposed theoretically [79]. Later, Cho et al. achieved it with simultaneous transmission (Sec. IV-B.1) by by-the-rules attackers [28]. The technique makes an initial collision with one of the victim's messages, which results in consecutive and repeated collisions with all of the victim's retransmission attempts until it becomes error-passive. Then, the attacker launches 19+ rounds of collisions each with a newly transmitted message by the victim until bus-off. Palanca et al. showed that bit-bangers could reduce this to a single round, by continuously attacking the victim's retransmissions until bus-off [80]. Later, CANOX showed that even by-the-rules attackers could do it in one round [24], [81], [82]. However, previous variations rely on attacking message retransmissions and could be slowed down by disabling them (see DA4). Recently, researchers proposed a single-round non-retransmission-based technique for bit-bangers [27], [83] called in-frame bus-off.

A5. Bus-Access Denial. By-the-rules attackers could deny other ECUs access to the bus [79], [35], [67] by flooding it with high-priority messages. An attacker may flood the bus with the highest-priority message to prevent all ECUs from accessing the bus, with a specific-priority message to selectively deny all messages with a lower priority. On the physical level, dominant voltage trumps recessive voltage. As a result, Bit-bangers could additionally: 1) inject a continuous dominant voltage to prevent any transmission [84], [35], 2) inject a dominant bit after EOF to cause an overload frame (Sec. II) then continuously

after every overload delimiter to prevent transmission without increasing error counters, or 3) deny arbitration for specific messages only by flipping a 1 to a 0 in the IDs [84], [26], [85].

A6. Synchronization Disruption. Bit-bangers could disrupt synchronization among different ECUs on the bus to cause various communication errors. Since CAN controllers use dynamic edge-based synchronization, attackers could exploit this ($\mathcal{V}_r$: ○) by injecting accurately-timed pulses within bits during frame transmission [84], [26], [72]. This leads ECUs to resynchronize based on the edges of those pulses and misread the beginnings and ends of different bits.

Controller Layer Takeaway: Most targeted, most diverse, and most replicable. The controller layer dominates attack interest, with 32 of 50 attack papers and 11 attack categories. Its attacks mostly exploit standard-rooted vulnerabilities, none rely on restricted ones, and most are easy to launch. Thus, 9 of 11 categories are highly replicable.

Target: Transwire Layer

A7. Signal Attenuation. Mohammed et al. [75] discovered that 8 bit-bangers ($\mathcal{D}_f$: ●) inducing transient signals near a receiver can attenuate the voltage signal between a transmitter and a receiver, possibly causing the message to be invisible to selected ECUs. This technique becomes more effective as the distance between the transmitter and the receiver grows ($\mathcal{V}_r$: ○).

Transwire Layer Takeaway: Least targeted and least diverse. The transwire layer appears in only 6 of 50 attack papers and spans 4 attack categories. All exploit standard-rooted vulnerabilities, but only one is highly replicable due to difficult attack steps or stronger attacker requirements.

Availability Takeaway: Frequently targeted and broadly replicable. Availability ranks second in attack papers (22 of 50), attack diversity (7 categories), and highly replicable attacks (4 of 7). All attacks exploit standard-rooted or widespread vulnerabilities; none rely on restricted vulnerabilities.

VI. CAN Defenses

In this section, we taxonomize CAN defenses as described in Sec. III. For each defense category, we assess its efficacy against relevant attacks and summarize the variants of the same defense approach reported in the literature, along with the challenges they face. Tbl. II summarizes all defenses and juxtaposes them against attacks to reveal defense gaps. The top row lists all attack categories. Highly replicable attacks are highlighted in light brown and light blue for by-the-rules attackers and bit-bangers, respectively. Each attack × defense cell reports the defense's assessment against that attack. A green cell marks an effective defense. A non-green cell means a defense can protect against an attack but not effectively. Letters in them list the criteria that the defense lacks, preventing it from being effective. We provide a spreadsheet at [12] detailing each assessment and its rationale.

Defense Assessment Criteria. We assess defense efficacy per attack. If a defense applies to multiple attacks, its assessment will vary across attacks based on the metrics below:

■ *Insurance ($\mathcal{I}_r$):* A defense insures *prevention (P)*, *detection (D)*, or *slowing (S)* by reducing its speed or effectiveness. *(*)* means the defense insures prevention only when combined with another approach.

■ *Coverage ($\mathcal{C}_v$):* Assesses how comprehensively a defense addresses different manifestations of an attack, including various attack paths or different attacker models. A defense is *restricted* (◐) when it only covers certain manifestations, leaving out known blind spots or attacker models. *Wide* coverage (●) means it addresses most or all variations and attackers.

■ *ECU Operational Efficiency ($\mathcal{E}_e$):* The efficiency is *low* (○) when it involves computationally expensive operations for every transmitted and received message. It is *moderate* (◐) if such operations are necessary only for some frequent messages or only outgoing ones. It is *high* (●) if they are needed only initially or sporadically with infrequent messages.

■ *Bus Performance ($\mathcal{B}_p$):* The performance is *low* (○) if it causes dropping or unschedulability (inability to put an upper bound on the maximum message delay), *moderate* (◐) if it is noticeable without causing message unschedulability, and *high* (●) if its busload or delay is barely noticeable.

■ *Infrastructure Compatibility ($\mathcal{I}_c$):* The compatibility is *low* if it requires significant hardware modifications to the bus or each ECU (○), *moderate* (◐) if it involves some extensive firmware updates or a combination of changes, and *high* (●) if it requires limited changes, including adding a monitor node or making small firmware/software updates.

■ *Validity ($\mathcal{V}_l$):* A defense might be *speculated* to work against an attack without being analyzed for CAN (○), proposed and *analyzed* for CAN systems but not evaluated (◐), or *evaluated* on CAN buses in real or test environments (●).

Effective Defense. We consider a defense effective if it offers at least *prevention* ($\mathcal{I}_r$), *wide* coverage ($\mathcal{C}_v$), *analyzed* validation ($\mathcal{V}_l$), and *moderate* ECU operation efficiency ($\mathcal{E}_e$), bus performance ($\mathcal{B}_p$), and infrastructure compatibility ($\mathcal{I}_c$).

Gaps: We define a defense gap as a highly replicable attack with no effective defense. Tbl. II highlights by-the-rules gaps in dark red, and bit-banger gaps in dark blue.

A. Confidentiality and Privacy Defenses

Protection: Higher Layer

DC1. Encryption and Obfuscation. Encryption can prevent communication surveillance (C1) with little infrastructure change ($\mathcal{I}_c$: ●). However, all cryptography-based approaches on CAN face 3 challenges: ECU performance, CAN message-length, and shared media key management. When encryption is used extensively, these challenges cause coverage, bus, and ECU efficiency issues. When used sporadically or outside normal operation (e.g., diagnostic sessions), they may not present a hurdle. Since these challenges are common to all cryptographic approaches, they are discussed in detail in Sec. VII-B. To overcome the length and performance challenges, researchers proposed the following:

■ *Message Length.* To get over CAN's 64-bit message length, researchers proposed using stream ciphers (e.g., RC4) [86],

TABLE II
CAN DEFENSE TAXONOMY AND EFFECTIVENESS ASSESSMENT PER ATTACK CATEGORY
HL: HIGHER LAYER, **CT**: CONTROLLER LAYER, **TW**: TRANSWIRE LAYER

Defenses \ Attacks			Confidentiality					Integrity										Availability						
			HL		CT	TW		HL			CT						TW	HL		CT				TW
			C1. Comm. Surveillance	C2. Memory Leakage	C3. Network Reconnaissance	C4. HW/Topology Profiling	C5. EM Emissions	I1. Fake Data Fabrication	I2. Mem./FW Manipulation	I3. Bypass Diag. Authentication	I4. Impersonation & Replay	I5. Frame Hijacking	I6. Double Receive	I7. Unorthodox Frames	I8. Poly-Semantic Frames	I9. Frame Tampering	I10. Physical FP Distortion	A1. ECU Func. Disabling	A2. Sybil Exhaustion	A3. Error Injection	A4. Error State Manipulation	A5. Bus Access Denial	A6. Synchronization Disruption	A7. Signal Attenuation
Conf.	HL	DC1. Encryption & Obfuscation	CEB																					
	CT	DC2. ID Anonymization			C							C					C			C	C			
	TW	DC3. Shielding & Slope Control					VI																	
Integrity	HL	DI1. Content Inspection						PC	P									P	P					
		DI2. Patching & Implementation																						
		DI3. ECU FW Hardening		C					C															
	CT	DI4. Message Authentication																						
		DI6. Victim Resistance																						
		DI7. Physical IDS														PC				PC	PC			PC
		DI8. Unify Sample Points													C	C								
	TW	DI9. Time & IFS Anonymization			C							C					C			C	C			
		DI10. Bit-Banger IDS											P			C				P	P	P	P	
		DI11. ECU Guardian							I		I	I	I	I	I	I	I	I	I	I	I			
Availability	CT	DA1. Error Handling IDS			P							P					PC				P			
		DA2. L2 Node Suspension																		C	C	C		
		DA3. ID-Priority Dissociation																		CBI	CBI	CBI		
		DA4. One-Shot Mode			PC							PC					PC				PC			
	TW	DA5. Dynamic Fragmentation										I	I	I	I	I	I			I	I	I	I	I

Color: Secure & Efficient, Secure, Efficient, Limited

Lacking Criteria: P No Prevention, C Restricted Coverage, V Limited Validation, E Low ECU Efficiency, B Low Bus Performance, I Low Infrastructure Compatibility

64-bit encryption [87], [88], truncating 128-bit block ciphers to 64 bits to generate pads and XORing them [14], or sending the payload over two messages [89], [90]. However, this increases the busload as it doubles each message ($\mathcal{B}_p$: ○) or forces it to adopt the maximum length ($\mathcal{B}_p$: ◐).

■ *Performance.* Some techniques attempt to reduce the overhead and busload impact ($\mathcal{E}_e$: ◐, $\mathcal{B}_p$: ●) by using non-encryption based obfuscation, such as Fisher–Yates shuffling [91]. However, this has limited effect with messages carrying single-value signals (e.g., all 0s). S2-CAN [92] proposed using both insertion and shuffling. Privacy-preserving transformations were proposed by the authors of SIFT [93], [38] for sensor data to hide privacy-revealing information, but they were never analyzed for CAN ($\mathcal{V}_l$: ○).

Protection: Controller Layer

DC2. ID Anonymization. Leaving the ID of a message as is, encryption (DC1) alone may not be able to completely preserve the confidentiality of a message and defend against confidentiality (C1) attacks. Similarly, lack of ID secrecy makes possible attacks that rely on simultaneous-transmission (C3, I5, I10, A3-A4), since they rely on sending a message with the same ID as a victim message simultaneously. However, since IDs play a pivotal role in priority, schedulability, and filtering, simply encrypting the ID field is problematic for CAN as it disrupts these aspects. To preserve priority and schedulability specifically, researchers proposed using priority bounds, pre-calculated tables, order-preserving one-way functions to anonymize the ID field [33], [94], [95], [96], [97]. However, earlier works [33] have shown several priority-preserving anonymization schemes reverse-engineerable. CanSafe [98] proposed hardware-level dissociation between IDs and priorities, but without analyzing its impact on schedulability ($\mathcal{I}_c$: ○, $\mathcal{V}_l$: ○). Other schemes split the ID into a fixed priority field and a dynamic field, but the strict correlation between the new priority field and old IDs makes them easily reverse-engineered. This limits their suitability to authentication and error handling attacks [99], [100], [70], but not confidentiality.

Protection: Transwire Layer

DC3. Shielding and Signal Slope Control. Little work has been done ($\mathcal{V}_l$: ○) to secure CAN from emission-based frame sniffing (C5). However, research suggests that shielding and slope control to make the transition edges between dominant and recessive voltages less sharp [101] may reduce emissions.

Confidentiality Takeaway: Weakly defended and restricted in coverage. No highly replicable confidentiality attack has an effective defense, and only one lower-replicability attack is effectively defended. Existing defenses have coverage issues, leaving known attack variants or attacker models unaddressed.

B. Integrity Defenses

Protection: Higher Layer

DI1. Content-Inspection. To detect messages containing fake or malicious data (I1, I4, and A1), researchers proposed inspecting message contents. The inspector is deployed as a central IDS (Fig. 5-①), although some are deployed with individual ECUs (Fig. 5-②). The inspected content may include

message size, ID, DLC, payload range, correlation with other messages and data sources, consistency with previous messages [102], or entropy [103], [104]. Such methods have been proposed for other buses as well [105], [106]. Inspection uses several machine learning algorithms [107], [108], [109], [110], [111], [112], [113], [114] sometimes in combination with rule-based checking [115]. Some inspectors are made specifically for higher layer protocols (e.g., J1939) [116]. Detecting whether the communicated signal value (e.g., of a sensor) is plausible often requires understanding the physical model of the entire system in which CAN is operating (e.g., a car). To do so, researchers proposed several approaches. Some works focused on using heterogeneous sensor consistency [117], [118]. Some attempted to understand the context or the state in which the vehicle operates [119], [120], [121], [122]. While these solutions achieved various levels of success, full system modeling remains a difficult problem with many variables ($\mathcal{C}_v$: ◐).

DI2. Better Implementation & Patching. Several attacks exploiting implementation problems could be prevented ($\mathcal{C}_v$: ●) by taking measures not impacting the ECU or bus operation. Checking sequence numbers for validity when handling multi-frame sessions [102] prevents attacks exploiting weak sequence number checking (C2, I2). Ignoring invalid protocol requests or timing out silent sessions [42] prevents attacks making invalid requests (A1) or establishing fake sessions (A2). Implementing strong diagnostic authentication by taking measures such as avoiding repetitive keys, using stronger authentication algorithms, and rate-limiting unsuccessful authentication attempts can prevent authentication bypass (I3) and other attacks that rely on it (C2, I2, A1). Verifying only authenticated devices could send ECU-disabling commands (A1) or higher-layer address claims (A2) protects availability.

DI3. ECU Firmware Hardening. These approaches prevent attacks targeting ECUs' firmware ($\mathcal{C}_v$: ●). Firmware obfuscation [123], [40] prevents attacks revealing firmware or memory (C2). Although not proposed or analyzed specifically for CAN, they are used in various systems with typically high computation [124], [125]. Firmware attestation [126], [127] or control flow integrity [128], [129] prevents attacks manipulating ECUs' firmware integrity or execution flow (I2). They often require extra computation ($\mathcal{E}_e$: ◐) and special hardware such as hardware security modules ($\mathcal{I}_c$: ◐). Many are not proposed or evaluated specifically for CAN ($\mathcal{V}_l$: ○→ ◐).

Higher Layer Takeaway: Defended in volume, exposed in priority. Most higher-layer attacks are effectively defended, but its highly replicable attacks are not.

Protection: Controller Layer

DI4. Message Authentication. To prevent impersonation and replay attacks (I4), cryptographic authentication faces challenges discussed in Sec. VII-B. To overcome these challenges, researchers proposed changing the authentication immediacy, authentication channel, and authentication responsibility:

■ *Authentication Channel.* Most solutions embed authentication and freshness data in the payload. If the transmitted data length is short, this would force the message to be longer.

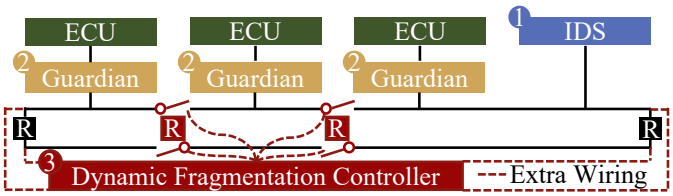

Fig. 5. Three defense types and their infrastructure needs.

However, since the maximum length of a CAN message is already short, this usually requires splitting the data and authentication information across several messages, increasing busload ($\mathcal{B}_p$: ○→ ◐) [88], [130], [131]. To tackle the message-length challenge, others suggested using unused message fields [92] or the ID [132]. Another channel is out-of-band signals or modified CAN fields using special equipment ($\mathcal{I}_c$: ○), such as the CRC [14]. CANAuth proposed using CAN+ [133], [134], a modified version of CAN requiring special controllers. CAN-HG [135] uses special hardware to alter the baud rate within the frame [136], allowing the embedding of authentication data. Timing channels are discussed separately in DI9.

■ *Authentication Immediacy. Immediate authentication* communicates the authentication information by the end of the message [137], [138], [83], [88], [130]. However, the short message length limits the amount of both payload and authentication data communicated in a message. To tackle the length challenge, *delayed authentication* waits for a subsequent message ($\mathcal{B}_p$: ○) to authenticate retrospectively [131]. An example is the TESLA protocol, which uses one-way chains [139]. Groza and Murvay developed and later refined a centralized version of it for CAN [140], [141].

■ *Authentication Responsibility.* In *per-ECU authentication*, each receiver authenticates incoming messages *($\mathcal{E}_e$: ○, $\mathcal{I}_r$: P, D)*. However, this may impact the ECU performance as ECUs are typically resource-constrained. In *centralized authentication*, a monitor node authenticates messages instead of every ECU *($\mathcal{I}_r$: D)*, eliminating the overhead of authenticating received messages *($\mathcal{E}_e$: ◐)*. The monitor is installed in a similar way to an IDS in Fig. 5-①, to prevent it from being a bottleneck. It has different keys for each ECU, alleviating the internal-attacker/group-key management issue (discussed in Sec. VII-B) ($\mathcal{C}_v$: ●). Nodes embed the authentication data in the authentication channel of outgoing messages [142], [137], [138], [143]. If immediate authentication is in use, the monitor could be used with frame destruction (DI5) [137], [138] to prevent malicious frames from being received since this approach has a very low false positive rate ($\mathcal{I}_r$: P*).

■ *Diagnostic Sessions.* Worth more consideration due to their range of control, the source, content, and freshness should be authenticated using per-ECU keys without group keys. To prevent obtaining authentication keys through brute-forcing or reverse engineering, OBD-II dongles limiting the number of attempts within a time window, stronger keys, or multi-factor authentication could be used. Assuming diagnostic sessions are infrequent and do not happen during normal operation, any

authentication channel could be used without impacting the performance of the bus or ECUs.

DI5. Frame Destruction. Injecting an error frame or overwriting a 1 with a 0 during frame transmission interrupts it and causes receiving CAN controllers to discard it without passing it to the higher layer. This technique must be combined with an IDS or a centralized authentication approach, enabling an IDS capable of injecting dominant bits to destroy the frames they deem malicious ($\mathcal{I}_r$: P*, $\mathcal{C}_v$: ●)[144], [83], [136], [137], [138].

DI6. Victim Resistance. Several works proposed that every ECU monitor whether they see their IDs transmitted on the bus by someone else and react upon detecting an impersonating message (I4) [145], [146], [147], [148]. The most effective approach is the secure transceiver. It is a dedicated hardware device that monitors the IDs autonomously without ECU software processing. It raises an error upon seeing an ID that belongs to its ECU, effectively employing frame destruction (DI5) but in a decentralized manner [145].

DI7. Physical IDS (PIDS). Monitoring signal characteristics may identify message source or transmitter locations. When compared with the expected source or location, it could identify anomalies or reveal the source ECU of an attack. The IDS is usually placed on the bus in parallel (Fig. 5-①) so it does not constitute a bottleneck. Accumulation of research led to solutions with low false-positive rates that may be combined with frame-destruction (DI5) to prevent impersonation & replay (I4). Researchers have proposed various IDS types that monitor physical signal characteristics:

■ *Voltage and Current-Based IDS (VIDS).* To detect impersonation and replay attacks, works profiled voltage characteristics of different nodes using a voltage sampling device [149], [48], [53], [49], [50], [150], [51], [52], [59], [58]. It compares a message's voltage profile to its legitimate transmitter to detect impersonation. Others extended VIDS to detect certain by-the-rules error handling attacks (A3-A4) [149]. Due to environmental variations, voltage profiles change over time and thus need to be updated with newly collected measurements. These online updates present fertile ground for training set poisoning attacks (i.e., attackers manipulating voltage measurements to slowly corrupt the profiles learned by the IDS) if they are not secured [70]. Others noticed that the electrical current drawn when sending a 1 bit by a single transmitter is different from that drawn when the bit was 0 but has been flipped by another attacker, and proposed using that to detect frame tampering attacks that flip 0 to 1 (I9) [76].

■ *Frequency and Time-Based IDS (TIDS).* Researchers noticed that the transmission frequency of various CAN messages with different IDs [151], [152], [153], [154], or the timing [44], [155], [156], [157] or sequence patterns [158], [159], [160] of periodic or request-response messages are largely fixed under normal operation and proposed monitoring them to detect anomalies. Others monitored timing features within the frame, such as bit-width, rise and fall times, and asymmetry [54], [55], [57], [56], [72] as they are largely fixed for the same transmitter.

■ *Attacker Localization.* Detecting transmitter (attacker) location is possible using wires connecting both ends of the bus to a TDC [61], or a voltage sampler [58], [59], [60]. Monitoring the difference in message arrival time between the two ends allows localizing the transmitter with an accuracy of up to 10 cm. These approaches work against all attacker types, including those with physical access.

DI8. Unifying Sampling Points. Unifying the voltage sample points of all controllers on the bus could prevent attacks that exploit sample point discrepancies to make different controllers read the same message differently. These include poly-semantic frames (I8) and frame tampering (I9) attacks.

Protection: Transwire Layer

DI9. Time and IFS Anonymization. Researchers proposed to encode secrets in timing channels (i.e., sending the message after a secret delay or at a secret time) to achieve authentication and protection against attacks targeting messages with predictable transmission times [161], [162], [163], [164]. However, because CAN's priority system can delay messages unpredictably, precisely controlling their absolute transmission time to encode secrets was not robust. These solutions also heavily impacted schedulability and incurred high false positive rates. ZBCAN [83] proposed eliminating the schedulability and false-positive issues using IFS (time since the last message terminated on the bus) instead of absolute message time as a robust channel to embed secrets in combination with frame destruction (DI5) to prevent impersonation. It can also work against error-handling attacks targeting messages with predictable transmission times (i.e., simultaneous-transmission-based: C3, I5, I10, and A3-A4) when launched by by-the-rules attackers, but it fails against bit-bangers.

DI10. Bit-Banger IDS. Since bit-banger attacks typically manipulate lower-layer events, they cannot be detected by monitoring message-level features (e.g., transmission frequency). Bit-banger IDSs monitor the link-layer at the bit level and physical layer features within frames to detect signatures that correspond to bit-banger attacks. This detects attacks including I4-I10 and A3-A6. ERACAN [72] proposed that a central IDS attach to the bus ($\mathcal{I}_c$: ●), watch it, and use frame destruction to prevent attacks such as I4-I5 and I7-I9. CANTXSec proposed that the IDS be connected to each ECU's CAN TX pins to identify the attacker among them ($\mathcal{I}_c$: ○) [165].

DI11. ECU Guardian. Researchers proposed physically installing a guardian between each ECU and the bus (Fig. 5-②, $\mathcal{I}_c$: ○). This could both protect the ECU from attacks from the bus and the bus from attacks from the ECU. This includes most attacks, including attacks on higher-layer protocols (e.g., J1939) [166]. In CAN-HG, the guardian could detect malicious activities itself or in collaboration with a central IDS. It can physically disconnect the ECU on detecting malicious activities by the guardian or receiving an out-of-band IDS signal [71], [136].

Integrity Takeaway: Strong attack–defense convergence. Integrity is both the most targeted CIA element and the most defended: 88 of 107 defense papers focus on it. This effectively defends 7 of its 10 attacks, including 4 of 6 highly replicable attacks, making integrity the only CIA element with effective defenses for highly replicable attacks.

C. Availability Defenses

Protection: Controller Layer

DA1. Error Handling IDS. Researchers proposed installing a central IDS that monitors errors on the bus and models the error counters of various ECUs to detect when they transition into the bus-off state [167]. They did not discuss other states such as the error-passive state, but the same method can likely detect it in the same way, thereby detecting any attack pushing the victim into specific error states (C3, I5, I10, and A4).

DA2. Layer-II Node Suspension. Since nodes stop communicating in the bus-off state, researchers proposed suspending by-the-rules attackers to the bus-off state to stop ongoing attacks. For attacks targeting availability, such as those attacking the messages of other ECUs or denying them bus access (A3-A5), this gives room for other nodes to communicate. The faster the attacker is suspended, the more effective the defense, especially for A5 (Several techniques explained in A4). However, it may be dangerous if the attacking ECU is safety-critical with only some malicious messages (e.g., if the brake ECU is attacking the infotainment ECU, it would not make sense to suspend the brake ECU). Further, it does not work against bit-bangers or down-to-earthers, as they do not abide by layer-II rules, including transitioning to the bus-off state.

DA3. ID-Priority Dissociation. Since the priority of accessing the bus in CAN is tied to the CAN ID, CANSafe proposed a complete change of CAN's physical layer ($\mathcal{I}_c$: ○) to dissociate ID and priority [98]. This could prevent attackers from flooding the bus with high-priority IDs to cause other message IDs to starve (A5). However, although priority affects schedulability, the authors did not analyze the impact of such a change on it.

DA4. One-Shot mode. Most CAN controllers offer disabling automatic retransmission (called one-shot transmission mode in several controllers) as an optional mode [168], [169]. This cheap defense does not require high operational or infrastructure costs to slow down or even prevent many by-the-rules ($\mathcal{C}_v$: ◐) attacks that rely on pushing the victim ECU to specific error states (C3, I5, I10, and A4). This is because many such attacks rely on repetitively attacking retransmitted messages to increase the victim ECU's error counter and push it into the desired error state. However, it has little effectiveness against non-retransmission-based attacks.

Observation: All 4 attacks targeting controller layer availability are highly replicable and have no effective defenses.

Controller Layer Takeaway: Heavily studied yet exposed. Although 70 of 107 defense papers address controller-layer attacks, most attacks still lack effective defenses. Existing solutions either lack prevention, especially against bit-bangers, or require high infrastructure changes.

Protection: Transwire Layer

DA5. Dynamic Fragmentation. Researchers suggested modifying the bus wiring to support attacker isolation, especially those targeting availability (e.g., A5) [170], [79]. CANARY installs a central IDS and relays at specific locations on the bus (Fig. 5-③, $\mathcal{I}_c$: ○). When the IDS detects an attack, CANARY uses the relays to disconnect each section of the bus one by one, search for the section where attacks are still observed to localize the attacker, and disconnect the section containing the attacker. However, this setup could result in isolating entire sections of the bus and not only the attacker. While its infrastructure cost and response time are relatively high, it offers the only line of defense against attacks with no other countermeasures and could be generalized to prevent any attack with an effective detection strategy ($\mathcal{I}_r$: P*).

Transwire Layer Takeaway: Covered where it matters. The transwire layer is the mirror image of the higher layer. It receives the least defense attention (17 of 107 defense papers) and has only one effective defense. Yet that defense covers the layer's only highly replicable attack (I10).

Availability Takeaway: Underdefended yet over-targeted. Availability is the second most targeted CIA element, but the least studied in defenses: only 5 of 107 papers focus on it. All its highly replicable bit-banger attacks lack effective defenses. Existing defenses either lack prevention or require high infrastructure changes.

VII. Root Cause Analysis

To determine whether CAN security problems are fundamentally CAN-specific, we analyze the root causes underlying its attacks and defense gaps. For each root cause, we determine whether it is *inherent* to CAN, *unique* to CAN, or both, and use these classifications to infer which attacks and gaps are themselves inherent or unique. This distinguishes problems rooted in CAN's design from those arising from broader shared-media properties, higher-layer protocols and implementations, or the surrounding system. Sec. VIII builds on these root causes to investigate whether the same problems are likely to persist in emerging IVBs.

Our analysis identifies 9 attack root causes ($\mathcal{R}$) and 3 defense-gap root causes ($\mathcal{DR}$). This section describes and analyzes each root cause along two non-mutually-exclusive qualities: *1) Inherency (I):* inherent to the standard or attacker model, rather than accidental or scenario-specific. *2) Uniqueness (U):* specific to the CAN family of standards. Because a root cause may underlie multiple attacks or gaps, we organize the analysis by root cause and identify the attacks or gaps each one underlies. Tbl. III summarizes the root-cause analysis. It maps each attack and defense gap to its underlying root causes and reports whether each root cause is inherent, unique, or both. A ✓ indicates that a root cause is indispensable to an attack or gap, while an ↑ indicates that addressing an attack root cause provides an effective defense. By-the-rules gaps are highlighted in red, bit-banger gaps in blue, and excluded attacks and root causes in gray.

A. Attack Root Causes ($\mathcal{R}$s)

■ *$\mathcal{R}$1: Attacker Access to Network (I).* The mere access of the attacker to the network, whether physically or remotely, is *inherent* to all of our attacker models (Sec. IV-A.1) and all attacks require it. While most attacks require additional root

TABLE III
ROOT CAUSES FOR CAN ATTACKS AND GAPS

	$\mathcal{R}1\,(I)$	$\mathcal{R}2\,(I)$	$\mathcal{R}3\,(I)$	$\mathcal{R}4\,(IU)$	$\mathcal{R}5\,(I)$	$\mathcal{R}6\,(IU)$	$\mathcal{R}7$	$\mathcal{R}8$	$\mathcal{R}9$	$DR1\,(I)$	$DR2\,(I)$	$DR3$	CAN FD	CAN XL (Err. On)	CAN XL (Err. Off)	10BASE-T1S	Ideal Bus
C1. Communication Surveil	✓	✓								✓			●	●	●	●	○
C2. Memory Leakage	✓						✓						●	●	●	●	●
C3. Network Reconnaissance	✓	✓	✓	✓	✓						✓		●	●	-	●	○
I1. Data Fabrication	✓											✓	●	●	●	●	●
I2. Mem & FW Manipulation	✓						✓						●	●	●	●	●
I3. Bypass Diagnostic Auth.	✓						✓			↑			●	●	●	●	●
I4. Impersonation & Replay	✓	✓								↑			●	●	●	●	○
I5. Frame Hijack	✓	✓	✓	✓	✓							↑	●	●	-	-	○
I10. Fingerprint Distortion	✓	✓	✓	✓	✓						↑		●	●	●	-	○
A1. ECU Disabling	✓						✓						●	●	●	●	●
A2. Sybil Exhaustion	✓						✓						●	●	●	●	●
A3. Error Injection	✓	✓	✓	✓	✓						✓		●	●	●	-	○
A4. Error State Manipulation	✓	✓	✓	✓	✓						✓		●	●	○	○	○
A5. Bus Access Denial	✓	✓	✓								✓		●	●	●	●	○
C4. HW & Topology Profile	✓	✓			✓				✓								
C5. EM Emission	✓				✓				✓								
I6. Double Receive	✓	✓		✓	✓			✓			✓						
I7. Unorthodox Frames	✓			✓				✓									
I8. Poly-Semantic Frames	✓	✓				✓		✓									
I9. Frame Tamper	✓	✓			✓			✓				↑					
A6. Synch. Disruption	✓	✓			✓			✓			✓						
A7. Signal Attenuation	✓	✓			✓			✓			↑						
CAN FD	●	●	●	●	●	●	●	-	-	◐	●	●					
CAN XL (Err. On)	●	●	●	●	●	●	●	-	-	◐	●	●					
CAN XL (Err. Off)	●	●	●	◐	●	●	●	-	-	◐	●	●					
10BASE-T1S	●	●	●	○	●	○	●	-	-	◐	●	●					
Ideal Bus	●	○	○	○	◐	○	●	-	-	○	○	●					

causes with it, it is enough for *fake data fabrication* (I1). It is not unique, as any comparison with other communication systems will assume it under any attacker model.

■ *$\mathcal{R}2$: Shared Media Properties (I).* Two inherent properties that are not unique to CAN, but common to all single-channel shared communication media and exploitable by all our attacker models, underlie many attacks. *1) Broadcast promiscuity:* this refers to the ability of a connected malicious node to anonymously monitor bus-wide activity and to have bus-wide reachability, allowing it to act on any observed frame, node, or the medium itself. This directly enables communication surveillance, network reconnaissance, and hardware and topology profiling attacks. It also underlies attacks that interfere with in-flight frames (I5, I10, and A7), attacks that induce errors or manipulate node state (A3 and A4), and attacks that exploit discrepancies among nodes sharing the same bus (I8, A6, I6, I4). *2) Mutual exclusion:* because transmission on a single shared channel prevents other nodes from transmitting at the same time, it gives an adversary leverage to deny others bus access or seize transmission opportunities, thereby underlying bus-access denial and frame hijacking.

■ *$\mathcal{R}3$: Exploitable Priority System (I).* As described in Sec. II, CAN uses a bus-wide priority mechanism, physically prioritizing dominant voltage over recessive. On the controller layer, this is utilized for collision resolution using arbitration, where lower CAN IDs (more leading dominant bits) have higher priority. This enables bus-access denial, whether by flooding the bus with low-CAN-ID frames or by continuously asserting dominant voltage. It also enables attacks that transmit past arbitration (network reconnaissance, frame hijacking, physical fingerprint distortion, and error injection and state manipulation), whether by matching the ID with victim messages to exploit the simultaneous-transmission technique for by-the-rules attackers (Sec. IV-B.1), or by directly flipping recessive voltage to dominant for bit-bangers. While this root-cause is inherent and exploitable by all attacker models, it is not unique as all shared media need priority and media access systems–many of which are exploitable, as shown in Sec. VIII.

■ *$\mathcal{R}4$: Error Handling Properties (IU).* Four inherent and unique properties of CAN's error handling mechanism are exploited by attackers: 1) it defines different criteria for frame validity between senders and receivers, allowing attackers to exploit this discrepancy (I6). 2) the types of errors and their impact, namely halting transmission and raising an error frame, as well as 3) the presence of behaviorally distinct error states, are exploited in attacks that inject errors to push victims into the attacker's desired error state and in attacks that exploit the different behavior of the error-passive state (C3, I5, I10, A3, and A4). 4) it leaves certain cases of frame non-conformity outside its covered error conditions, allowing attackers to transmit malformed frames without provoking a uniform standard response (I7). Paradoxically, some of these properties are used for defense. Specifically, without frame destruction (DI5) as an impact of an active error, impersonation and replay would be exposed as gaps (Tbl. II). Similarly, thanks to error-states and error-types, fast layer-II node-suspension (DA2) is an effective bus-access denial defense, without which it would be exposed as a by-the-rules gap.

■ *$\mathcal{R}5$: Physical Signaling (I).* Several characteristics of CAN physical signaling are exploitable by attackers ranging from by-the-rules attackers to down-to-earthers: 1) despite differential voltage and twisted wires, CAN is still vulnerable to emitting EMI (C5) due to lack of shielding and relatively high switching speed. 2) minor electrical differences among nodes using the bus as a shared electrical medium allow hardware and topology profiling (C4) and physical fingerprint distortion (I10). 3) using push-pull signaling to drive dominant bits and high-impedance for recessive bits enables attacks that rely on bit-flips and pulses (C3, I5, I6, I9, A3, A4, A7). 4) dynamic edge synchronization allows bit-bangers manipulating edges to disrupt synchronization among nodes (A6). These features are inherent but not unique. As will be shown in Sec. VIII, many other protocols' physical signaling share them.

■ *$\mathcal{R}6$: CAN Bit Sampling (IU).* To translate voltage levels into bit values, CAN nodes sense the voltage at specific times called sample points whose timing is configurable, not the same across nodes, and happens once per bit. Exploiting discrepancies between node sample points allows attackers to insert signal transitions between them to send messages that have differing interpretations (I8). This is inherent and unique as other protocols do not combine these sample-point qualities, particularly configurability.

■ *$\mathcal{R}7$: Vulnerable Higher-Layer Protocols and Implementation.*

Higher-layer protocols and their implementations can introduce exploitable logic and software weaknesses. Vulnerable features, such as request-response exchanges, session mechanisms, and functionality-disabling commands, can be abused by by-the-rules attackers to launch ECU functionality disabling (A1), and sybil exhaustion (A2) attacks. At the implementation level, software flaws such as memory-corruption bugs and weaknesses in diagnostic-security handling can be exploited for privileged information exposure and internal-ECU-manipulation attacks (C2, I2, and I3). This is not inherent as the CAN standard does not specify or mandate any implementations or higher-layer protocols. It is also not unique, as these protocols and implementations could be used with other communication media.

■ *$\mathcal{R}$8: Layer-II Rule Violation.* The ability of bit-bangers to bypass CAN's controller layer rules, placing and reading arbitrary signals on the bus, underlies all bit-banger attacks (I6-I9, and A6-A7). It also underlies the bit-banger variants of several attacks that could be launched using different methods for by-the-rules attackers (C3, I5, I10, and A3-A5). By definition, this is not inherent to CAN as it does not stem from its rules but from the violation of them, jeopardizing other parts of the standard. It is either an ECU design problem allowing for the bypass or manipulation of the CAN controller, or an access problem, allowing physical installation of non-CAN-conforming devices (Sec. IV-A.3).

■ *$\mathcal{R}$9: Layer-I Rule Bypass.* Down-to-earthers' ability to bypass CAN's physical-layer rules through privileged physical access and specialized equipment allows them to inject or collect physical signal measurements in ways CAN was not designed for. This, along with other root causes, underlies down-to-earther attacks such as hardware and topology profiling (C4) and capturing electromagnetic emissions (C5).

B. Defense Gap Root Causes ($\mathcal{DR}$s)

■ *$\mathcal{DR}$1: Cryptography Challenges (I).* Cryptography on CAN faces three challenges. First, *performance*: commercial ECUs often have limited processing budgets, making cryptographic operations costly if applied to all outgoing and incoming messages. Second, *message length*: CAN's 64-bit maximum payload complicates carrying authentication data. For example, AUTOSAR suggests using 24 authentication bits. If a message carries more than 40 bits of data, little room remains for authentication. If it carries less, authentication data can significantly increase message length and busload. The same length constraint also makes conventional block-cipher encryption cumbersome, since block sizes often exceed the message size. Third, and more fundamentally, CAN faces an inherent but not unique *shared-media key-management dilemma*. On a broadcast medium, group keys [14], [90], [86] preserve efficiency and can protect against external attackers (i.e., an ECU without the group key). However, communication is exposed to internal attackers: a compromised ECU that holds the group key can decrypt all traffic and impersonate any group member. Asymmetric cryptography for general communication is often impractical on resource-constrained ECUs. Pairwise keys between each sender-receiver pair [90], [86], [130] reduce that exposure, but do not fit the broadcast nature of CAN and may impose substantial communication overhead, since the same message may need to be sent multiple times under different keys. While workarounds have been proposed to address performance and message length, little has been proposed to solve this shared-media key-management dilemma, the main culprit behind the internal attacker issue. For key establishment, some works rely on a central server [90], [86], [171], [172], while others propose mechanisms such as identity-based signatures [173], implicit certificates [172], or schemes based on transwire or controller-layer characteristics [174], [175], [90]. Yet many of these approaches still assume pre-shared secrets, which become particularly dangerous when combined with group-key settings, since compromising one node jeopardizes the entire group. Finally, although some key-exchange protocols could in principle support revocation by re-running the exchange, few CAN works directly address internal-attacker revocation [171], [176], and reported revocation times remain comparable to key-exchange or long-term-key update times.

Overall, fixing the performance and message length challenges mainly improves bus or ECU efficiency, whereas addressing the shared-media key-management dilemma fills the communication surveillance gap (C1) and provides effective defenses against bypassing diagnostic authentication (I3) and impersonation and replay (I4).

■ *$\mathcal{DR}$2: Costly Attacker Signal Filtering and Disabling (I).* On a broadcast bus, availability attacks aimed at disrupting communication (A3-A6) can only be stopped by filtering the attacker's signals or disabling the attacker node, so other nodes have room to communicate. Similarly, attacks that rely on injecting malicious signals into specific parts of a message (C3, I6) can only be prevented this way. The lack of a cheap and effective attacker filtering and disabling approach is the fundamental reason for the bit-banger gaps of C3, I6, A3-A6, where defenses cannot simultaneously achieve prevention for all attacker models and good infrastructure compatibility. Layer-II node suspension (DA2) is a cheap form of node disabling but works only against by-the-rules attackers, leaving the bus vulnerable to other attacker models. Addressing stronger attackers by adding guardians (DI11) or fragmentation mechanisms (DA5) requires expensive infrastructure changes. Combined with the existing detection approaches, addressing this challenge closes these gaps and provides an effective defense against integrity and availability attacks against the transwire layer (I10, A7).

■ *$\mathcal{DR}$3: Difficult Environment Understanding.* Determining whether a message is malicious may be infeasible without understanding the bigger system (DI1). For example, if an attacker compromises the brake ECU and sends fake brake values without impersonation, the only way to detect this fabrication is to understand how these values relate to the bigger system (e.g., the vehicle), which may involve physical modeling. This $\mathcal{DR}$ is not inherent to CAN but to its environment (e.g., vehicle) and could be solved for open or less complex environments (e.g., an elevator). It is also not unique, as it will be present in other communication media operating within the same environment. Addressing this root-cause closes the fake data fabrication gap

(I1) and effectively detects tampering with in-flight frames (I5, and I9), which could be turned into prevention to provide an effective defense when combined with frame destruction (DI5).

C. Observations and Takeaways

■ *On CAN 2.0 and Root Causes.* Most root causes are not unique to CAN. Specifically, only vulnerable *error handling* ($\mathcal{R}4$) and *bit sampling* ($\mathcal{R}6$) are unique to the CAN family. Interestingly, no defense-gap root cause is unique to CAN.

CAN fundamental security problems stem from shared single-channel media properties, vehicular constraints, and ECU implementation weaknesses that could recur in any shared-media system under similar conditions.

■ *On CAN 2.0 and Attacks and Gaps.* An attack or a gap usually depends on multiple root causes. If any required root cause is not inherent, the attack is not inherent. Conversely, if any of the root causes underlying an attack or a gap is unique, it is unique. Thus, although 6/9 attack root causes are inherent, only 9/22 attacks are inherent, 8/22 attacks are unique, and only 5/22 are both inherent and unique. Relative to gaps, only one is inherent, and it is inherent to shared-media communication, not CAN. The 4 unique gaps are unique only because their underlying attacks rely on CAN-unique root causes, not because the absence of effective defense is unique. Most importantly, no gap is both inherent and unique.

VIII. Transferability Analysis

In this section, we assess whether CAN security problems are likely to persist in other in-vehicle buses (IVBs). Building on the root causes identified in Sec. VII, we investigate whether the same root causes appear in alternative IVBs and extrapolate whether the corresponding CAN attacks and gaps are likely to carry over. The premise is that if an IVB inherits the root-cause pattern of an attack or gap, then the attack or gap is likely to transfer. Our investigation is therefore a targeted standard review centered on the identified root causes, rather than a broad security review of each IVB. Where standard review is insufficient to determine the presence of a root cause, we may use formal verification, as with the optional standardized link-layer security mechanisms CANsec and MACsec.

Tbl. III summarizes the transferability analysis. IVB rows report root-cause presence, where (●) means largely present, (◐) somewhat present, and (○) largely absent. IVB columns report the resulting attack/gap transferability, where (●) means likely, (○) unlikely, and (−) unresolved. Excluded attacks and root causes are shown in gray.

Emerging IVB Selection. Although point-to-point [177], [178] and switched communication technologies [179], [180] are appearing in vehicles, they require substantially more wiring. The wiring harness is already among the vehicle's most costly components and its third heaviest after the engine and chassis [181], [182]. We therefore expect wiring cost and weight to confine such technologies to essential use cases, with buses continuing to dominate in-vehicle communication. Among the many existing choices, we focus on three representative in-vehicle bus technologies that are strong candidates to replace classic CAN [183], [184]: CAN FD, CAN XL, and Automotive Ethernet (10BASE-T1S). We select them because manufacturers are most likely to adopt successors built on already established automotive communication ecosystems, namely CAN and Ethernet. To separate security problems rooted in bus protocols from those rooted in the surrounding system, we further consider a hypothetical ideal bus that provides secure logical channelization with confidentiality, integrity, and availability.

Investigation Methodology. To ensure a fair cross-IVB comparison, we focus only on weaknesses rooted in the standard. Accordingly, we adopt a by-the-rules attacker model and exclude attacker models that depend on violating, bypassing, or otherwise stepping outside the standard rules. We also assume that vehicle higher-layer protocols running on top of CAN could run on top of any of the considered IVBs. Under these comparison rules, $\mathcal{R}1$ and $\mathcal{R}7$ are satisfied for all IVBs, while $\mathcal{R}8$ and $\mathcal{R}9$, together with the attacks that rely on them, are excluded from the comparison.

A. Transferability Analysis

1. CAN FD. Reviewing the standard [185], [186], [187], CAN FD is a shared single-channel broadcast bus and thus shares the same universal shared media properties ($\mathcal{R}2$). It uses the same priority system as CAN 2.0 and its priority is settable for by-the-rules attackers ($\mathcal{R}3$). Despite slight modifications, its error-handling mechanism retains the same different definitions of frame validity, error types and impacts, behaviorally distinct states, and corner cases ($\mathcal{R}4$). It uses the same physical signaling with higher switching speeds than CAN 2.0 and without stipulating shielding as a requirement. Similar to CAN 2.0, only dominant bits are driven by push-pull. Nodes perform dynamic edge synchronization ($\mathcal{R}5$). Bits are sampled by reading the voltage level once at configurable sample points ($\mathcal{R}6$). It takes no measures to allow selective attacker signal filtering ($\mathcal{DR}2$). While its longer payload alleviates a great part of the performance and length challenges outlined in $\mathcal{DR}1$, it is still subject to the same shared media key-management dilemma.

2. CAN XL. Reviewing the standard [186], [187], CAN XL is a shared single-channel broadcast bus and shares the same universal shared media properties ($\mathcal{R}2$), enabling communication surveillance (C1) and impersonation and replay (I4). It uses a similar priority system but conventional ID functionalities are split into a Priority ID and an Acceptance Field, both settable by by-the-rules attackers ($\mathcal{R}3$), making bus-access denial (A5) likely.

Error handling could be enabled or disabled by by-the-rules attackers. **When enabled**, it is similar to CAN 2.0, suggesting equivalents of all error-injection-based attacks (A3, A4, C3, I5, I10) are possible. **When disabled**, considerable changes exist, including removing behaviorally distinct error states ($\mathcal{R}4$). This likely avoids error state manipulation (A4). However, this also removes node suspension (DA2) as a possible defense, as a by-the-rules attacker can disable its own error handling so it is never suspended into the bus-off state, regardless of the victim's setting. Since node suspension is an effective defense against many attacks, and the only effective defense

against bus-access denial (A5), it is possibly exposed as a gap for both error handling settings. Both 0s and 1s are driven using push-pull. This still makes equivalents of error injection likely. However, the outcome of a 0-1 collision will be less predictable than CAN 2.0, where it would always be 0. Therefore, the presence of frame hijacking (I5) and network reconnaissance (C3) equivalents cannot be determined and needs further research.

It supports higher switching speeds and, similar to CAN 2.0, does not mandate shielding ($\mathcal{R}5$), uses dynamic edge synchronization, with bits read once at configurable sample points ($\mathcal{R}6$). It is compatible with CANsec [188] and its length is significantly longer, alleviating performance and length challenges. To assess if it addresses the internal attacker issue ($\mathcal{DR}1$), we formally verified CANsec (App. A). Our verification shows that as the current in-development version of CANsec stands, it provides authenticity, freshness, and confidentiality only against external attackers but not internal attackers ($\mathcal{DR}1$) and hence does not close any open defense gaps.

3. 10BASE-T1S. Reviewing the standard [189], 10BASE-T1S is a shared half-duplex broadcast bus, with the same universal shared media properties ($\mathcal{R}2$), making equivalents of communication surveillance (C1), and impersonation and replay (I4) possible. Priority is managed by Physical Layer Collision Avoidance (PLCA). Each node's priority is determined by its local ID. Similar to CAN, it is settable for by-the-rules attackers ($\mathcal{R}3$), making bus-access denial (A5) possible.

It has different error handling rules and no error states, likely avoiding error state manipulation (A4). However, it lacks features that could be used as effective defenses, such as frame destruction (DI5) or node suspension (DA2), which in turn is likely to expose impersonation and replay (I4) and bus access denial (A5) as gaps. Network reconnaissance (C3) is likely, not using errors, but using inherent properties (e.g., node ID). Research on whether it harbors properties allowing equivalents of error injection (A3), frame hijacking (I5), and fingerprint distortion (I10) is needed.

Similar to CAN, it uses differential signaling with dynamic edge synchronization, does not stipulate shielding, and supports higher switching speeds than CAN 2.0. It uses differential Manchester encoding, where both 0s and 1s are driven with push-pull. This makes outcomes of 0/1 collisions unpredictable, unlike CAN, where it is always 0 ($\mathcal{R}5$). Unlike CAN, receivers interpret bits by reading voltage transitions at fixed start and middle of bits, without configurable sample points ($\mathcal{R}6$).

Like CAN, it takes no measures to facilitate selective attacker signal filtering ($\mathcal{DR}2$). It is natively compatible with MACsec, and its maximum length is significantly longer, alleviating the performance and length challenges. To assess if it addresses the internal attacker issue ($\mathcal{DR}1$), we formally verified MACsec (App. A). Our verification shows that as the 10BASE-T1S version of MACsec stands, it provides authenticity, freshness, and confidentiality only against external attackers but not internal attackers ($\mathcal{DR}1$) and hence does not close any open defense gaps.

4. Hypothetical Ideal Bus. To distinguish between security

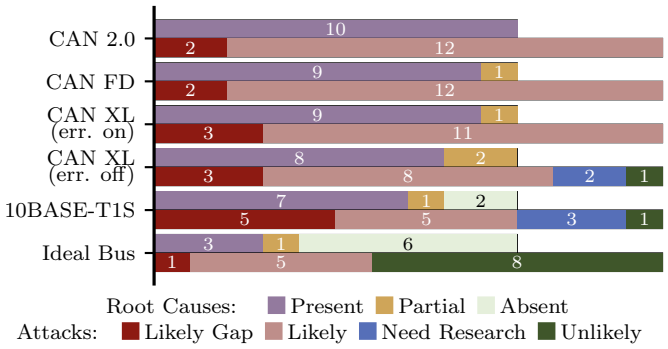

Fig. 6. Transferability of CAN 2.0 security problems to emerging IVBs.

problems that can, in principle, be eliminated by bus protocol design and those that can only be addressed by securing the system surrounding the bus, we assume the presence of an ideal bus. Note these are not meant to be practical recommendations on how a more secure bus should be designed, but establish an upper bound on the security guarantees achievable by bus protocol design alone. The bus offers secure logical channelization. Each channel consists of one authorized transmitter, one or more authorized receivers, and a defined set of messages. Transmission from different channels may occur simultaneously on the same physical medium (using Code Division Multiple Access CDMA or other methods). Nodes outside a channel cannot observe messages inside a channel or learn their contents (for example, using VLAN-like technologies). This eliminates $\mathcal{R}2$. A message can only be transmitted by the authorized transmitter and its priority cannot be tampered with (eliminating $\mathcal{R}3$). Nodes' transmission opportunities are guaranteed and no single node can deny all other nodes from bus access (using scheduling, or other methods). All these rules are enforced by protocol controllers attached to each node. The controller cannot be bypassed or tampered with by a by-the-rules attacker, and bus access can be revoked from compromised nodes (addressing $\mathcal{DR}2$). Nodes in a channel share a channel-specific key. Messages are protected using fast authenticated encryption. If a compromised channel member is detected, a fast revocation mechanism removes it from the channel and securely distributes new keys to the remaining members (eliminating $\mathcal{DR}1$). CAN-specific error handling ($\mathcal{R}4$) and bit sampling ($\mathcal{R}6$) properties are eliminated (by adopting mechanisms different from CAN). $\mathcal{R}5$ is partially addressed by physical layer design. The bus may avoid push-pull signaling, mandate shielding to reduce EMI, and use a shared clock to avoid dynamic edge synchronization. However, transmitters will still exhibit distinguishable physical signal characteristics due to hardware differences.

Even with such an ideal hypothetical bus, $\mathcal{R}1$, $\mathcal{R}7$, and $\mathcal{DR}3$ will remain, reproducing attacks C2, I1-I3, A1-A2 and the fake data fabrication gap. These causes are rooted in the system surrounding the bus, not the bus itself. They will reproduce no matter how perfect the bus is. Specifically, appropriate control is needed to restrict physical access or remote exploitation of

vulnerabilities in bus-connected devices. Otherwise, an attacker can still access an ideal secure bus ($\mathcal{R}1$). Insecure features and implementations of higher layer protocols ($\mathcal{R}7$), and more accurately modeling the cyber-physical environment ($\mathcal{DR}3$), are outside the scope of what the physical and data link layer specifications of a bus can secure.

B. Observations and Takeaways

■ *On IVBs and Root Causes.* Fig. 6 summarizes the results of the transferability analysis. As shown, **CAN FD** shares all compared CAN root causes, making all considered attacks and gaps likely transferable. **CAN XL** has two error-handling settings. When enabled, it fully shares 9 root causes, and partially shares 1. As such, 14 attacks and 2 gaps under comparison are likely transferable, and 1 additional attack is likely exposed as a gap. When disabled, it fully shares 8 root causes, and partially shares 2. Consequently, 11 attacks and both gaps are likely transferable. 1 attack is unlikely to transfer, and 2 require further research to determine. 1 additional attack is likely exposed as a gap. **10BASE-T1S** fully shares 7 root causes, partially shares 1, and avoids 2. As such, 10 attacks are likely transferable, 3 need further research, and 1 is unlikely to transfer. Both gaps are likely transferable with 3 additional attacks likely exposed as gaps. Finally, an **ideal secure bus** fully shares 3 root causes, partially shares 1, and avoids 6. 8 attacks may be avoided and 6 remain, where 1 remains as a gap. This shows that certain problems are rooted in the system surrounding the bus and will reproduce no matter how perfect the bus is unless a new approach to vehicle security is taken, where the entire system is considered as one whole.

- Emerging IVBs do not solve CAN's fundamental security problems. Most CAN security problems likely transfer.
- CAN is more securable than perceived. Each examined IVB inherits at least as many transferable attacks with defense gaps as CAN, before accounting for IVB-specific risks.
- Certain CAN security problems can only be solved by securing the system surrounding the bus.

■ *Limitation.* The findings of the transferability analysis were only indicative. The attacks identified as likely transferable were not validated on the respective buses. Such validation could be performed by formally analyzing parts of each bus protocol to determine attack viability, or empirically by setting up a testbed. Of the 14 compared attacks, 5 relied on higher-layer protocols. These attacks would need to be evaluated not only on the respective buses, but with the higher-layer protocols also set up on the ECUs. For CAN variations (CAN FD and CAN XL), most techniques used in CAN 2.0 could be directly utilized. However, for 10BASE-T1S, some techniques would need to be adapted to the specifics of the protocol. For example, instead of achieving bus access denial by faking a high-priority CAN ID, an attacker may need to fake the node ID. Across all buses, attack success could then be measured individually for each attack using the same or adapted metrics the attack authors used (e.g., swiftness and suppression rates for error-state manipulation/bus-off attacks, message loss and delay for bus-access denial attacks, etc.).

IX. Future Directions

A major finding of our study is that many CAN security problems are rooted in causes common to single-channel shared communication media and vehicular environments, rather than in CAN-specific design choices. Accordingly, we believe the following research directions (Fig. 7) are beneficial not only for CAN 2.0, but also for any standard that may replace it.

Insider-Aware IVB Cryptography. Future IVB cryptography must treat compromised ECUs as first-class adversaries, not just external attackers. IVB cryptography has focused on fitting authentication and confidentiality into constrained buses: limited payloads, strict timing, low ECU compute budgets, and bus-load sensitivity. Newer IVBs reduce some barriers through longer frames, higher bandwidth, and hardware-assisted line-rate protection. However, our analysis shows that a deeper problem behind several defense gaps is shared-media key management: a compromised ECU with valid group keys can still decrypt traffic or impersonate messages. Future work should therefore focus on insider-aware IVB cryptography, fast revocation, secure rekeying, continuous attestation, and zero trust architecture.

Shared-Media Filtering and Isolation. One finding of this study is that, whatever replaces CAN 2.0, future shared-media IVBs need a practical way to contain a compromised node without taking down the whole channel ($\mathcal{DR}2$). IDSs can detect problems, and encryption may preserve confidentiality, but neither helps if the attacker can still occupy the medium, inject errors, disturb timing, force retransmissions, or manipulate physical/data-link-layer behavior. In a single-channel bus, one malicious transmitter can affect every receiver, while the network often lacks a cheap way to suppress only the malicious frame, dominant bit, error, timing disturbance, or node. Without such containment, many CAN gaps will transfer to successor buses, as shown in Sec. VIII. Existing CAN approaches, such as ECU guardians (DI11) and dynamic fragmentation (DA5), attempt this isolation, but their infrastructure cost limits adoption. Future work should therefore develop low-cost shared-media filtering and isolation mechanisms that selectively block messages or signals, disconnect malicious nodes, or isolate bus segments while preserving normal operation.

Cyber-Physical State Consistency. Many IVB attacks are not mere identity-forgery attacks, but semantic lies about the physical state. This is why fake data fabrication remains a difficult gap to close in CAN and the other IVBs we investigated: a forged speed or brake value may be well-formatted, correctly timed, and even authenticated if sent by a compromised brake ECU. Detecting such attacks requires determining whether a message is physically and operationally plausible in the broader cyber-physical context. Future research should therefore move toward cyber-physical state consistency checking: determining whether a bus message is plausible given driver input, sensor readings, actuator behavior, ECU state, timing, control logic, and vehicle dynamics. Research should focus on capturing both expected message-level behavior and

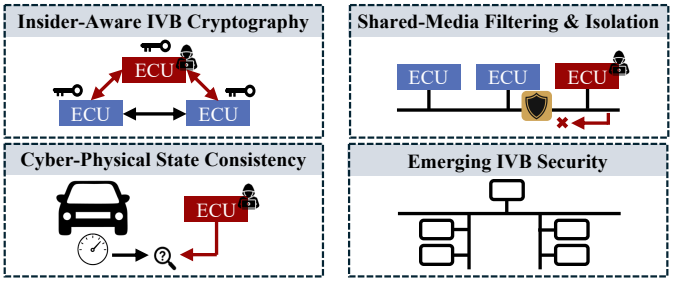

Fig. 7. Research directions for future in-vehicle communication security.

the physical consequences those messages should produce across ECUs and vehicle subsystems. Digital twins could provide the foundation for this direction. However, challenges regarding proprietary designs, incomplete knowledge, runtime uncertainty, and vehicle-specific behavior are expected to arise.

Emerging IVB Security. One finding of this study is that newer IVBs likely inherit many of CAN's security problems. Higher bandwidth, longer payloads, revised frame formats, and optional security mechanisms can improve important constraints, but they do not remove issues arising from shared access, broadcast communication, priority mechanisms, physical signaling, or compromised insiders. However, our investigation only establishes root-cause presence and uses it to infer the likelihood of transferability. Future research should therefore validate these indications by determining whether the corresponding attacks and gaps are practically exploitable, how they unfold in each IVB, and what countermeasures can stop them. More importantly, research should not be limited to CAN-transferable problems: each emerging IVB may introduce protocol-specific vulnerabilities through its own frame format, media-access rules, encoding, error behavior, and security extensions. A recent formal analysis of CAN XL validates several of our transferability predictions, showing that CAN XL retains known CAN 2.0 MAC-layer vulnerabilities and their corresponding attacks, while also uncovering new CAN XL-specific vulnerabilities beyond our comparison [190]. This work is time-sensitive: once an IVB becomes widely deployed, structural weaknesses become difficult and expensive to correct.

X. Conclusions

In this paper, we addressed a critical gap in CAN security literature: the lack of a coherent systematization framework. Through an extensive literature survey, we introduced a comprehensive taxonomy and assessment models to articulate attacker privileges, the attacks they enable, their targets, replicability, defense efficacy, and impact. Leveraging this framework, we highlighted where the literature is concentrated or lacking and identified CAN's defense gaps. We then analyzed the root causes of CAN insecurity to determine whether they are inherent or unique to CAN, and conducted a follow-up investigation to assess whether future IVB technologies decisively resolve conventional CAN security challenges. Our findings challenge common perceptions: CAN is more securable than often assumed, most of its fundamental security problems stem from shared single-channel communication properties or vehicular constraints, and most of these problems are transferable to other IVBs. Finally, we concluded that securing CAN and future IVBs requires addressing shared root causes more than CAN-specific causes, and outlined four key research directions to secure IVB systems down the road.

XI. Acknowledgments

We thank the anonymous reviewers for their comments and suggestions. This work was supported in part by the Office of Naval Research (ONR) under Grant N00014-18-1-2674, the National Science Foundation (NSF) under Grants CNS-2145744 and CNS-214464, and the National Center for Transportation Cybersecurity and Resiliency (TraCR) under Award Number 2589-211-2026310. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the ONR, NSF, or TraCR, and the U.S. Government assumes no liability for the contents or use thereof.

XII. Ethics Considerations

As a systematization of knowledge paper, this study did not involve manipulating real-world systems. Our formal verification explored the protection boundary of CANsec/MACsec and whether it extends beyond current cryptographic CAN solutions. It did not identify previously unknown protocol vulnerabilities. Nevertheless, we have been in contact with CAN in Automation (CiA). They approved our request to access the in-development CANsec draft standard and conduct this research, and have been informed of our results. They asked to emphasize that CANsec and CKA are under development, which we have done. All works referenced in this paper are publicly accessible. No private or confidential information was used.

References

[1] Horizon Connect, "Why CAN bus isn't enough for modern vehicles," https://tinyurl.com/4upfuvd3s.
[2] Embedded, "Ditching the CAN bus," https://tinyurl.com/3jff4b8x.
[3] H. J. Jo and W. Choi, "A survey of attacks on controller area networks and corresponding countermeasures," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
[4] S. T. F. T. Loto'aniu, "A survey of attacks over controller area networks and potential countermeasures," Ph.D. dissertation, Auckland University of Technology, 2021.
[5] O. Avatefipour and H. Malik, "State-of-the-art survey on in-vehicle network communication (CAN-bus) security and vulnerabilities," *arXiv preprint arXiv:1802.01725*, 2018.
[6] R. S. Rathore, C. Hewage, O. Kaiwartya, and J. Lloret, "In-vehicle communication cyber security: challenges and solutions," *Sensors*, no. 17, 2022.
[7] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of can bus security challenges," *Sensors*, no. 8, 2020.
[8] E. Aliwa, O. Rana, C. Perera, and P. Burnap, "Cyberattacks and countermeasures for in-vehicle networks," *ACM Computing Surveys*, 2021.
[9] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, G. Madzudzo, and M. Cheah, "AI-based intrusion detection systems for in-vehicle networks: A survey," *ACM Computing Surveys*, 2023.
[10] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, "A survey of intrusion detection for in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, no. 3, 2019.

[11] A. Lotto, F. Marchiori, A. Brighente, and M. Conti, "A survey and comparative analysis of security properties of can authentication protocols," *arXiv preprint arXiv:2401.10736*, 2024.

[12] "SoK artifacts," https://doi.org/10.5281/zenodo.22555092.

[13] S. Mazloom, M. Rezaeirad, A. Hunter, and D. McCoy, "A security analysis of an in-vehicle infotainment and app platform," in *USENIX Workshop on Offensive Technologies*, 2016.

[14] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle CAN," *IEEE Transactions on Intelligent Transportation Systems*, 2014.

[15] H. Wen, Q. A. Chen, and Z. Lin, "Plug-N-pwned: Comprehensive vulnerability analysis of OBD-II dongles as a new over-the-air attack surface in automotive IoT," in *USENIX Security Symposium*, 2020.

[16] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, 2015.

[17] ——, "A survey of remote automotive attack surfaces," *Black Hat USA*, 2014.

[18] S. Checkoway, D. Mccoy, B. Kantor *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *USENIX Security Symposium*, 2011.

[19] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking Tesla from wireless to CAN bus," *Black Hat USA*, 2017.

[20] S. Nie, L. Liu, Y. Du, and W. Zhang, "Over-the-air: How we remotely compromised the gateway, BCM, and autopilot ECUs of Tesla cars," *Black Hat USA*, 2018.

[21] T. Telegraph, "Thieves are hacking into cars through their headlights, experts warn," https://tinyurl.com/fw3cy8cc, 2023.

[22] D. Gessner, M. Barranco, A. Ballesteros, and J. Proenza, "Designing sfiCAN: a star-based physical fault injector for CAN," in *International Conference on Emerging Technologies and Factory Automation*, 2011.

[23] V. I. GmbH, "User manual canstress, version 2.1." 2006.

[24] K. Serag, R. Bhatia, V. Kumar, Z. B. Celik, and D. Xu, "Exposing new vulnerabilities of error handling mechanism in CAN," in *USENIX Security Symposium*, 2021.

[25] S. Kulandaivel, S. Jain, J. Guajardo, and V. Sekar, "Cannon: Reliable and stealthy remote shutdown attacks via unaltered automotive microcontrollers," in *IEEE Symposium on Security and Privacy*, 2021.

[26] A. de Faveri Tron, S. Longari, M. Carminati, M. Polino, and S. Zanero, "Canflict: Exploiting peripheral conflicts for data-link layer attacks on automotive networks," in *ACM SIGSAC Conference on Computer and Communications Security*, 2022.

[27] S. Fröschle and A. Stühring, "Analyzing the capabilities of the CAN attacker," in *European Symposium on Research in Computer Security*, 2017.

[28] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[29] L. Yu, Y. Liu, P. Jing, X. Luo, L. Xue, K. Zhao, Y. Zhou, T. Wang, G. Gu, S. Nie *et al.*, "Towards automatically reverse engineering vehicle diagnostic protocols," in *USENIX Security Symposium*, 2022.

[30] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown CAN bus networks," *Vehicular Communications*, 2017.

[31] M. D. Pesé, T. Stacer, C. A. Campos *et al.*, "LibreCAN: Automated CAN message translator," in *ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[32] M. Marchetti and D. Stabili, "READ: Reverse engineering of automotive data frames," *IEEE Transactions on Information Forensics and Security*, no. 4, 2019.

[33] M. Lukasiewycz, P. Mundhenk, and S. Steinhorst, "Security-aware obfuscated priority assignment for automotive can platforms," *ACM Transactions on Design Automation of Electronic Systems*, no. 2, 2016.

[34] D. Frassinelli, S. Park, and S. Nürnberger, "I know where you parked last summer: Automated reverse engineering and privacy analysis of modern cars," in *IEEE Symposium on Security and Privacy*, 2020.

[35] K. Koscher, A. Czeskis, F. Roesner *et al.*, "Experimental security analysis of a modern automobile," in *IEEE Symposium on Security and Privacy*, 2010.

[36] H. Wen, Q. Zhao, Q. A. Chen, and Z. Lin, "Automated cross-platform reverse engineering of CAN bus commands from mobile apps," in *Network and Distributed System Security Symposium*, 2020.

[37] Comma.ai, "opendbc: Democratize access to car decoder rings," https://github.com/commaai/opendbc, 2019.

[38] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile driver fingerprinting." in *Privacy Enhancing Technologies Symposium*, 2016.

[39] U. Fugiglando, P. Santi, S. Milardo, K. Abida, and C. Ratti, "Characterizing the "driver dna" through CAN bus data analysis," in *ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services*, 2017.

[40] A. Milburn, N. Timmers, N. Wiersma, R. Pareja, and S. Cordoba, "There will be glitches: Extracting and analyzing automotive firmware efficiently," *Black Hat USA*, 2018.

[41] J. Van den Herrewegen, "Automotive firmware extraction and analysis techniques," Ph.D. dissertation, University of Birmingham, 2021.

[42] R. Chatterjee, S. Mukherjee, and J. Daily, "Exploiting transport protocol vulnerabilities in SAE J1939 networks," in *Symposium on Vehicle Security and Privacy*, 2023.

[43] J. Shin, H. Kim, S. Lee, W. Choi, D. H. Lee, and H. J. Jo, "RIDAS: Real-time identification of attack sources on controller area networks," in *USENIX Security Symposium*, 2023.

[44] K. T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *USENIX Security Symposium*, 2016.

[45] S. Kulandaivel, T. Goyal, A. K. Agrawal, and V. Sekar, "Canvas: Fast and inexpensive automotive network mapping," in *USENIX Security Symposium*, 2019.

[46] S. U. Sagong, X. Ying, A. Clark *et al.*, "Cloaking the clock: Emulating clock skew in controller area networks," in *ACM/IEEE International Conference on Cyber-Physical Systems*, 2018.

[47] X. Ying, S. U. Sagong, A. Clark, L. Bushnell, and R. Poovendran, "Shape of the cloak: Formal analysis of clock skew-based intrusion detection system in controller area networks," *IEEE Transactions on Information Forensics and Security*, 2019.

[48] P.-S. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Processing Letters*, no. 4, 2014.

[49] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ECUs using inimitable characteristics of signals in controller area networks," *IEEE Transactions on Vehicular Technology*, 2018.

[50] M. Kneib, O. Schell, and C. Huth, "EASI: Edge-based sender identification on resource-constrained platforms for automotive networks," in *Network and Distributed System Security Symposium*, 2020.

[51] M. Kneib and C. Huth, "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks," in *ACM SIGSAC Conference on Computer and Communications Security*, 2018.

[52] M. Foruhandeh, Y. Man, R. Gerdes *et al.*, "SIMPLE: Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks," in *Annual Computer Security Applications Conference*, 2019.

[53] K. T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[54] J. Zhou, P. Joshi, H. Zeng, and R. Li, "Btmonitor: Bit-time-based intrusion detection and attacker identification in controller area network," *ACM Transactions on Embedded Computing Systems*, no. 6, 2019.

[55] S. Ohira, A. K. Desta, T. Kitagawa, I. Arai, and F. Kazutoshi, "Divider: Delay-time based sender identification in automotive networks," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference*. IEEE, 2020.

[56] O. Schell, C. Oechsler, and M. Kneib, "Asymmetric symbol and skew sender identification for automotive networks," *IEEE Transactions on Information Forensics and Security*, 2022.

[57] S. Ohira, A. K. Desta, I. Arai, and K. Fujikawa, "PLI-TDC: Super fine delay-time based physical-layer identification with time-to-digital converter for in-vehicle networks," in *ACM Asia Conference on Computer and Communications Security*, 2021.

[58] E. Levy, A. Shabtai, B. Groza, P.-S. Murvay, and Y. Elovici, "CAN-LOC: Spoofing detection and physical intrusion localization on an in-vehicle CAN bus based on deep features of voltage signals," *IEEE Transactions on Information Forensics and Security*, 2023.

[59] P.-S. Murvay and B. Groza, "TIDAL-CAN: Differential timing based intrusion detection and localization for controller area network," *IEEE Access*, 2020.

[60] B. Groza, P.-S. Murvay, L. Popa, and C. Jichici, "CAN-SQUARE-decimeter level localization of electronic control units on CAN buses," in *European Symposium on Research in Computer Security*, 2021.

[61] M. Roeschlin, G. Camurati, P. Brunner, S. Mridula, and C. Srdjan, "EdgeTDC: On the security of time difference of arrival measurements

in CAN bus systems," in *Network and Distributed System Security Symposium*, 2023.
[62] O. Schell and M. Kneib, "SPARTA: signal propagation-based attack recognition and threat avoidance for automotive networks," in *ACM Asia Conference on Computer and Communications Security*, 2023.
[63] Y. X. Teo, J. Chen, N. Ash, A. R. Ruddle, and A. J. Martin, "Forensic analysis of automotive controller area network emissions for problem resolution," in *2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium*. IEEE, 2021.
[64] E. Evenchick, "An introduction to the CANard toolkit," in *Blackhat Conference*, 2015.
[65] ——, "CANtact-the open source car tool," https://hackaday.io/project/579-cantact, 2018.
[66] S. Mukherjee, H. Shirazi, I. Ray, J. Daily, and R. Gamble, "Practical DoS attacks on embedded networks in commercial vehicles," in *Information Systems Security International Conference*, 2016.
[67] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *Def Con*, 2013.
[68] T. Lauser, G. Munoz Molto, and C. Krauß, "(un) authenticated diagnostic services: A practical evaluation of vulnerabilities in the uds authentication service," in *Cyber Security in CarS Workshop*, 2024.
[69] Y. Burakova, B. Hass, L. Millar, and A. Weimerskirch, "Truck hacking: An experimental analysis of the SAE J1939 standard." in *USENIX Workshop on Offensive Technologies*, 2016.
[70] R. Bhatia, V. Kumar, K. Serag, Z. B. Celik, M. Payer, and D. Xu, "Evading voltage-based intrusion detection on automotive can," in *Network and Distributed System Security Symposium*, 2021.
[71] Ken Tindell, "Can bus security: Attacks on can bus and their mitigations."
[72] Z. Tang, K. Serag, S. Zonouz, Z. B. Celik, D. Xu, and R. Beyah, "ERACAN: Defending against an emerging can threat model," *ACM SIGSAC Conference on Computer and Communications Security*, 2024.
[73] Ken Tindell, "Janus attack."
[74] L. Yue, Z. Li, T. Yin, and C. Zhang, "CANCloak: Deceiving two ecus with one frame," in *Workshop on Automotive and Autonomous Vehicle Security*, no. 02, 2021.
[75] A. Z. Mohammed, Y. Man, R. Gerdes, M. Li, and Z. B. Celik, "Physical layer data manipulation attacks on the can bus," in *Intl. Workshop on Automotive and Autonomous Vehicle Security*, 2022.
[76] M. Rogers, P. Weigand, J. Happa, and K. Rasmussen, "Detecting CAN attacks on J1939 and NMEA 2000 networks," *IEEE Transactions on Dependable and Secure Computing*, 2022.
[77] Y. Lee, S. Woo, J. Lee, Y. Song, H. Moon, and D. H. Lee, "Enhanced android app-repackaging attack on in-vehicle network," *Wireless Communications and Mobile Computing*, 2019.
[78] S. Kumar, I. Karim, E. Bertino, and A. Arora, "Security attacks to the name management protocol in vehicular networks," in *ISOC Symposium on Vehicle Security and Privacy*, 2024.
[79] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*, 2004.
[80] A. Palanca, E. Evenchick, F. Maggi, and S. Zanero, "A stealth, selective, link-layer denial-of-service attack against automotive networks," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2017.
[81] K. Serag, V. Kumar, Z. B. Celik, R. Bhatia, M. Payer, and D. Xu, "Attacks on CAN error handling mechanism," in *International Workshop on Automotive and Autonomous Vehicle Security*, 2022.
[82] K. Serag, "Proactive vulnerability identification and defense construction – the case for can," Ph.D. dissertation, 2023.
[83] K. Serag, R. Bhatia, A. Faqih, M. Ozgur Ozmen, V. Kumar, Z. B. Celik, and D. Xu, "ZBCAN: A zero-byte CAN defense system," in *USENIX Security Symposium*, 2023.
[84] P.-S. Murvay and B. Groza, "DoS attacks on controller area networks by fault injections from the software layer," in *International Conference on Availability, Reliability and Security*, 2017.
[85] A. Humayed, F. Li, J. Lin, and B. Luo, "CANSentry: Securing CAN-based cyber-physical systems against denial and spoofing attacks," in *European Symposium on Research in Computer Security*, 2020.
[86] Z. Lu, Q. Wang, X. Chen, G. Qu, Y. Lyu, and Z. Liu, "LEAP: A lightweight encryption and authentication protocol for in-vehicle communications," in *IEEE Intelligent Transportation Systems Conference*, 2019.
[87] M. Jukl and J. Čupera, "Using of tiny encryption algorithm in CAN-bus communication," *Research in Agricultural Engineering*, 2016.
[88] G. Bella, P. Biondi, G. Costantino, and I. Matteucci, "Toucan: A protocol to secure controller area network," in *ACM Workshop on Automotive Cybersecurity*, 2019.
[89] P. Mundhenk, A. Paverd, A. Mrowca, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Security in automotive networks: Lightweight authentication and authorization," *ACM Transactions on Design Automation of Electronic Systems*, 2017.
[90] O. Pfeiffer, "Implementing scalable CAN security with cancrypt," *Embedded Systems Academy*, 2017.
[91] J. Yeom and S. Seo, "A methodology of CAN communication encryption using a shuffling algorithm," in *International Conference on Connected and Autonomous Driving*, 2020.
[92] M. D. Pesé, J. W. Schauer, J. Li, and K. G. Shin, "S2-can: Sufficiently secure controller area network," in *Annual Computer Security Applications Conference*, 2021.
[93] M. Enev, J. Jung, L. Bo, X. Ren, and T. Kohno, "Sensorsift: balancing sensor data privacy and utility in automated face understanding," in *Annual Computer Security Applications Conference*, 2012.
[94] B. Groza, L. Popa, and P.-S. Murvay, "Highly efficient authentication for can by identifier reallocation with ordered cmacs," *IEEE Transactions on Vehicular Technology*, 2020.
[95] R. Brown, A. Marti, C. Jenkins, and S. Shannigrahi, "Dynamic address validation array (DAVA) a moving target defense protocol for can bus," in *ACM Workshop on Moving Target Defense*, 2020.
[96] K. Cheng, Y. Bai, Y. Zhou, Y. Tang, D. Sanan, and Y. Liu, "CANeleon: Protecting CAN bus with frame ID chameleon," *IEEE Transactions on Vehicular technology*, no. 7, 2020.
[97] W. Wu, R. Kurachi, G. Zeng, Y. Matsubara, H. Takada, R. Li, and K. Li, "IDH-CAN: A hardware-based ID hopping CAN mechanism with enhanced security for automotive real-time applications," *IEEE Access*, 2018.
[98] A. Roy and S. K. Madria, "Cansafe: An mtd based approach for providing resiliency against dos attack within in-vehicle networks," in *25th ITSC*, 2022.
[99] S. Woo, D. Moon, T.-Y. Youn, Y. Lee, and Y. Kim, "Can id shuffling technique (cist): Moving target defense strategy for protecting in-vehicle can," *IEEE Access*, 2019.
[100] K. Han, A. Weimerskirch, and K. G. Shin, "A practical solution to achieve real-time performance in the automotive network by randomizing frame identifier," *ESCAR*, 2015.
[101] F. Silva, M. Quilez, J. Llop, X. Torres, and P. Riu, "EMI from an automotive CAN bus," in *International Symposium on Electromagnetic Compatability.*, 1999.
[102] M. Müter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in *International Conference on Information Assurance and Security*, 2010.
[103] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *IEEE Intelligent Vehicles Symposium*, 2011.
[104] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *IEEE International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow*, 2016.
[105] J. D. Eckhardt, T. E. Donofrio, and K. Serag, "System and method of monitoring data traffic on a MIL-STD-1553 data bus," 2019, uS Patent 10,467,174.
[106] ——, "Bus data monitor," 2020, uS Patent 10,691,573.
[107] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *IEEE International Conference on Data Science and Advanced Analytics*. IEEE, 2016.
[108] S. Longari, D. H. N. Valcarcel, M. Zago, M. Carminati, and S. Zanero, "CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network," *IEEE Transactions on Network and Service Management*, no. 2, 2020.
[109] M. H. Shahriar, Y. Xiao, P. Moriano, W. Lou, and Y. T. Hou, "CANShield: Signal-based intrusion detection for controller area networks," *arXiv preprint arXiv:2205.01306*, 2022.
[110] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS one*, no. 6, 2016.
[111] ——, "A novel intrusion detection method using deep neural network for in-vehicle network security," in *IEEE Vehicular Technology Conference*, 2016.

[112] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, 2020.

[113] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-based intrusion detection system for in-vehicle CAN bus communications," *IEEE Access*, 2020.

[114] M. H. Shahriar, W. Lou, and Y. T. Hou, "CANtropy: Time series feature extraction-based intrusion detection systems for controller area networks," in *Internation Symposium on Vehicle Security and Privacy*, 2023.

[115] S. Tariq, S. Lee, H. K. Kim, and S. S. Woo, "CAN-ADF: The controller area network attack detection framework," *Computers & Security*, 2020.

[116] S. Mukherjee, J. Walkery, I. Rayz, and J. Daily, "A precedence graph-based approach to detect message injection attacks in J1939 based networks," in *Annual Conference on Privacy, Security and Trust*, 2017.

[117] A. Ganesan, J. Rao, and K. Shin, "Exploiting consistency among heterogeneous sensors for vehicle anomaly detection," SAE Technical Paper, Tech. Rep., 2017.

[118] F. Guo, Z. Wang, S. Du, H. Li, H. Zhu, Q. Pei, Z. Cao, and J. Zhao, "Detecting vehicle anomaly in the edge via sensor consistency and frequency characteristic," *IEEE Transactions on Vehicular Technology*, no. 6, 2019.

[119] C. Y. Chen, "Context-aware detection and resolution of data anomalies for semi-autonomous cyber-physical systems," Ph.D. dissertation, 2022.

[120] A. Wasicek, M. D. Pesé, A. Weimerskirch, Y. Burakova, and K. Singh, "Context-aware intrusion detection in automotive control systems," in *ESCAR USA*, 2017.

[121] C.-Y. Chen, K. G. Shin, and S. Dadras, "Context-aware anomaly detection using vehicle dynamics," in *The 27th International Symposium on Research in Attacks, Intrusions and Defenses*, 2024.

[122] L. Xue, Y. Liu, T. Li, K. Zhao, J. Li, L. Yu, X. Luo, Y. Zhou, and G. Gu, "SAID: State-aware defense against injection attacks on in-vehicle network," in *USENIX Security Symposium*, 2022.

[123] L. Yu, J. Deng, R. R. Brooks, and S. B. Yun, "Automobile ECU design to avoid data tampering," in *Annual Cyber and Information Security Research Conference*, 2015.

[124] B. Cyr, J. Mahmod, and U. Guin, "Low-cost and secure firmware obfuscation method for protecting electronic systems from cloning," *IEEE Internet of Things Journal*, no. 2, 2019.

[125] E. Stefanov, M. v. Dijk, E. Shi, T.-H. H. Chan, C. Fletcher, L. Ren, X. Yu, and S. Devadas, "Path ORAM: an extremely simple oblivious RAM protocol," *Journal of the ACM*, 2018.

[126] J. Van Bulck, J. T. Mühlberg, and F. Piessens, "VulCAN: Efficient component authentication and software isolation for automotive control networks," in *Annual Computer Security Applications Conference*, 2017.

[127] M. Khodari, A. Rawat, M. Asplund, and A. Gurtov, "Decentralized firmware attestation for in-vehicle networks," in *Cyber-Physical System Security Workshop*, 2019.

[128] P. Nasahl, R. Schilling, and S. Mangard, "Protecting indirect branches against fault attacks using arm pointer authentication," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2021.

[129] Y. Wang, C. L. Mack, X. Tan, N. Zhang, Z. Zhao, S. Baruah, and B. C. Ward, "InsectACIDE: Debugger-based holistic asynchronous CFI for embedded system," in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2024.

[130] O. Hartkopp, C. Reuber, and R. Schilling, "MaCAN-message authenticated CAN," in *ESCAR*, 2012.

[131] S. Nürnberger and C. Rossow, "–vatiCAN–vetted, authenticated can bus," in *International Conference on Cryptographic Hardware and Embedded Systems*, 2016.

[132] A.-I. Radu and F. D. Garcia, "LeiA: A lightweight authentication protocol for CAN," in *European Symposium on Research in Computer Security*, 2016.

[133] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus," in *ECRYPT Workshop on Lightweight Cryptography*, 2011.

[134] T. Ziermann, S. Wildermann, and J. Teich, "CAN+: A new backward-compatible controller area network (CAN) protocol with up to 16× higher data rates." in *Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 2009.

[135] Canis Automotive Labs, "CAN-HG overview: Augmenting classic CAN for performance and security, white paper," https://canislabs.com/downloads/1905-2020-12-14-CAN-HG-overview.pdf.

[136] K. Tindell, "Defending in-vehicle CAN buses from attacks," in *The Ground Vehicle Systems Engineering and Technology Symposium*, 2022.

[137] R. Kurachi, Y. Matsubara, H. Takada, N. Adachi, Y. Miyashita, and S. Horihata, "CaCAN-centralized authentication system in CAN (controller area network)," in *ESCAR*, 2014.

[138] E. Wang, W. Xu, S. Sastry, S. Liu, and K. Zeng, "Hardware module-based message authentication in intra-vehicle networks," in *International Conference on Cyber-Physical Systems*, 2017.

[139] A. Perrig, J. Tygar, A. Perrig, and J. Tygar, "TESLA broadcast authentication," *Secure Broadcast Communication: In Wired and Wireless Networks*, 2003.

[140] B. Groza and P.-S. Murvay, "Higher layer authentication for broadcast in controller area networks," in *International Conference on Security and Cryptography*, 2011.

[141] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Transactions on Industrial Informatics*, no. 4, 2013.

[142] H. J. Jo, J. H. Kim, H.-Y. Choi, W. Choi, D. H. Lee, and I. Lee, "MAuth-CAN: Masquerade-attack-proof authentication for in-vehicle networks," *IEEE Transactions on Vehicular Technology*, 2020.

[143] E. Wagner, F. Basels, J. Bauer, T. Zimmermann, K. Wehrle, and M. Henze, "Caiba: Multicast source authentication for CAN through reactive bit flipping," in *IEEE European Symposium on Security and Privacy*, 2025.

[144] H. Giannopoulos, "CAN stomper: Active bus level countermeasures for the controller area network," Ph.D. dissertation, 2015.

[145] N. Semiconductors, "NXP tja115x secure can transceiver family," https://www.nxp.com/docs/en/fact-sheet/SECURCANTRLFUS.pdf,.

[146] T. Matsumoto, M. Hata, M. Tanabe, K. Yoshioka, and K. Oishi, "A method of preventing unauthorized data transmission in controller area network," in *Vehicular Technology Conference*, 2012.

[147] T. Dagan and A. Wool, "Parrot, a software-only anti-spoofing defense system for the CAN bus," *ESCAR EUROPE*, 2016.

[148] M. D. Pesé, B. Gozubuyuk, E. Andrechek, H. Olufowobi, M. Hamad, and K. G. Shin, "MichiCAN: Spoofing and denial-of-service protection using integrated CAN controllers," in *Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2025.

[149] W. Choi, K. Joo, H. J. Jo *et al.*, "VoltageIDS: Low-level communication characteristics for automotive intrusion detection system," *IEEE Transactions on Information Forensics and Security*, 2018.

[150] Y. Xun, Y. Zhao, and J. Liu, "VehicleEIDS: A novel external intrusion detection system based on vehicle voltage signals," *IEEE Internet of Things Journal*, no. 3, 2021.

[151] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *World Congress on Industrial Control Systems Security*, 2015.

[152] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," in *International Conference on Information Networking*, 2016.

[153] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks–practical examples and selected short-term countermeasures," *Reliability Engineering & System Safety*, 2011.

[154] C. Young, H. Olufowobi, G. Bloom, and J. Zambreno, "Automotive intrusion detection based on constant CAN message frequencies across vehicle driving modes," in *ACM Workshop on Automotive Cybersecurity*, 2019.

[155] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "SAIDuCANT: Specification-based automotive intrusion detection using controller area network (CAN) timing," *IEEE Transactions on Vehicular Technology*, no. 2, 2019.

[156] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *IEEE Annual Conference on Privacy, Security and Trust*, 2017.

[157] Z. Tang, K. Serag, S. Zonouz, Z. B. Celik, D. Xu, and R. Beyah, "WIP: Intrusion detection and localization for CAN by extracting propagation delay features from message intervals," in *USENIX Symposium on Vehicle Security and Privacy*, 2025.

[158] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences," in *IEEE Intelligent Vehicles Symposium*, 2017.

[159] A. Tomlinson, J. Bryans, S. A. Shaikh, and H. K. Kalutarage, "Detection of automotive CAN cyber-attacks by identifying packet timing anomalies in time windows," in *International Conference on Dependable Systems and Networks Workshops*, 2018.

[160] S. Katragadda, P. J. Darby, A. Roche, and R. Gottumukkala, "Detecting low-rate replay-based injection attacks on in-vehicle networks," *IEEE Access*, 2020.
[161] B. Groza, L. Popa, and P.-S. Murvay, "Incanta-intrusion detection in controller area networks with time-covert authentication," in *Security and Safety Interplay of Intelligent Software Systems*, 2018.
[162] ——, "CANTO-covert authentication with timing channels over optimized traffic flows for CAN," *IEEE Transactions on Information Forensics and Security*, 2020.
[163] X. Ying, G. Bernieri, M. Conti, and R. Poovendran, "TACAN: Transmitter authentication through covert channels in controller area networks," in *ACM/IEEE International Conference on Cyber-Physical Systems*, 2019.
[164] S. Vanderhallen, J. Van Bulck, F. Piessens, and J. T. Mühlberg, "Robust authentication for automotive control networks through covert channels," *Computer Networks*, 2021.
[165] D. Donadel, K. Balasubramanian, A. Brighente, B. Ramasubramanian, M. Conti, and R. Poovendran, "CANTXSec: A deterministic intrusion detection and prevention system for can bus monitoring ecu activations," in *International Conference on Applied Cryptography and Network Security*, 2025.
[166] J. Daily, D. Nnaji, and B. Ettlinger, "Securing CAN traffic on J1939 networks," in *Workshop on Automotive and Autonomous Vehicle Security*, 2021.
[167] S. Longari, M. Penco, M. Carminati, and S. Zanero, "Copycan: An error-handling protocol based intrusion detection system for controller area network," in *ACM Workshop on Cyber-Physical Systems Security & Privacy*, 2019.
[168] Microchip, "MCP2515: Stand-alone CAN controller with SPI," http://ww1.microchip.com/downloads/en/devicedoc/21801e.pdf.
[169] Philips, "SJA1000: Stand-alone CAN controller," https://www.nxp.com/docs/en/data-sheet/SJA1000.pdf, 2000.
[170] B. Groza, L. Popa, P.-S. Murvay, Y. Elovici, and A. Shabtai, "CANARY-a reactive defense mechanism for controller area networks based on active relays," in *USENIX Security Symposium*, 2021.
[171] A. Musuroi, B. Groza, L. Popa, and P.-S. Murvay, "Fast and efficient group key exchange in controller area networks (CAN)," *IEEE Transactions on Vehicular Technology*, 2021.
[172] D. Püllen, N. A. Anagnostopoulos, T. Arul, and S. Katzenbeisser, "Using implicit certification to efficiently establish authenticated group keys for in-vehicle networks," in *IEEE Vehicular Networking Conference*, 2019.
[173] B. Groza and P.-S. Murvay, "Identity-based key exchange on in-vehicle networks: CAN-FD & FlexRay," *Sensors*, 2019.
[174] A. Mueller and T. Lothspeich, "Plug-and-secure communication for CAN," *CAN Newsletter*, 2015.
[175] S. Jain and J. Guajardo, "Physical layer group key agreement for automotive controller area networks," in *International Conference on Cryptographic Hardware and Embedded Systems (CHES)*, 2016.
[176] Z. Wu, J. Zhao, Y. Zhu, K. Lu, and F. Shi, "Research on in-vehicle key management system under upcoming vehicle network architecture," *Electronics*, no. 9, 2019.
[177] M. Alliance, "MIPI A-PHY Long-reach SerDes," *v1.1*.
[178] "ASA Overview and a Proof-of-Concept ASA Transceiver."
[179] I. S. for Ethernet, "Amendment 1: Physical Layer Specifications and Management Parameters for 100 Mb/s Operation over a Single Balanced Twisted Pair Cable (100BASE-T1)," *IEEE Computer Society*.
[180] ——, "Amendment 4: Physical Layer Specifications and Management Parameters for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable," *IEEE Computer Society*.
[181] Mobility Foresights, "Global electric vehicle wiring harness market," https://tinyurl.com/3rmv7pfe.
[182] Markets and Markets, "Automotive wiring harness market," https://tinyurl.com/4v9seydu.
[183] Robert Bosch GmbH, "Technical comparison: CAN XL vs. 10BASE-T1S," https://tinyurl.com/yc3un22z, 2022.
[184] U. Schaefer, "The art of networking (series 4): CAN XL and 10BASE-T1S – dataline performance of two new actors in the networking arena," https://tinyurl.com/5dnt4xb6, 2023, renesas Electronics.
[185] Bosch GmbH, R, "Can with flexible data-rate specification," 2012.
[186] *Road Vehicles — Controller area network (CAN). ISO-11898-1:2024*. International Organization for Standardization (ISO).
[187] *Road Vehicles — Controller area network (CAN). ISO-11898-2:2024*. International Organization for Standardization (ISO).
[188] *CiA 613-2: CAN XL add-on services*.
[189] I. S. for Ethernet, "Amendment 5: Physical Layer Specifications and Management Parameters for 10 Mb/s Operation and Associated Power Delivery over a Single Balanced Pair of Conductors," *IEEE Computer Society*.
[190] Z. Tang, K. Serag, Z. B. Celik, V. Ganesh, S. Zonouz, and R. Beyah, "A formal security analysis of {CAN}{XL}," in *35th USENIX Security Symposium (USENIX Security 26)*, 2026, pp. 2227–2246.
[191] "IEEE standard for local and metropolitan area networks-media access control (MAC) security," *IEEE Std 802.1AE-2018*.
[192] "IEEE standard for local and metropolitan area networks-port-based network access control," *IEEE Std 802.1X-2020*.
[193] A. Zeh and K. W. Tindell, "Secure communications using pre-shared keys," 2025, uS Patent App. 18/508,443.
[194] B. Blanchet *et al.*, "Modeling and verifying security protocols with the applied pi calculus and ProVerif," *Foundations and Trends® in Privacy and Security*, 2016.
[195] Open Alliance, "Automotive MKA specification," https://opensig.org/wp-content/uploads/2025/03/OA_MACsec_Automotive-MKA-v1.pdf, 2025.
[196] AUTOSAR, "Specification of MACsec key agreement," https://www.autosar.org/fileadmin/standards/R24-11/CP/AUTOSAR_CP_SWS_MACsecKeyAgreement.pdf, 2024.

APPENDIX A
CANSEC / MACSEC FORMAL VERIFICATION

As a supplementary check within the IVB root-cause analysis, we examine whether CANsec and automotive MACsec close the internal-attacker gap ($\mathcal{DR}1$). In their automotive configurations, both organize nodes into Connectivity Associations (CAs) whose members share a long-term symmetric Connectivity Association Key (CAK) to authenticate then establish short-term Secure Association Keys (SAK) to protect communication. However, a shared CAK alone does not determine insider protection because the protocols and their key-agreement mechanisms could still introduce per-sender or pairwise keys, node-specific derivation, or compromised-member exclusion and rekeying. We therefore model the protocols and the key-management configurations specified or currently proposed for automotive use and verify their security properties against both external and internal attackers. We find that none of the modeled configurations introduces key or privilege separation among members of the same CA, or other mechanisms that could close $\mathcal{DR}1$. Since MACsec supports other key-management configurations and CANsec remains under development, we additionally test whether idealized revocation used in conjunction with these protocols could close the gap. The verification code and additional details are available at [12].

MACsec. MACsec is a link-layer security add-on compatible with all Ethernet variants that provides message authenticity, freshness, and confidentiality [191]. Nodes form a CA. MKA [192] authenticates CA members that hold a CAK and includes a key-server role. As shown in Fig. 8, the shared CAK is used to derive a Key Encrypting Key (KEK) and Integrity Check Key (ICK). The key server generates SAKs, encrypts them using the KEK, and authenticates their distribution using the ICK. CA members derive the same KEK and ICK from the shared CAK, obtain the distributed SAKs, and use them to protect regular MACsec communication.

CANsec. CANsec is an *in-development* link-layer security add-on for CAN XL modeled after MACsec [188]. Its current

proposal similarly uses CAs, CAKs, and SAKs, with CANsec Key Agreement (CKA) establishing SAKs. CKA is *still in early development*, but two approaches are currently being considered: 1) an approach modeled after MKA and 2) In-line Key Agreement (IKA) [193]. In IKA, SAKs are derived using the CAK, an incrementing Key Number (KN), and a Key Derivation Function (KDF), as shown in Fig. 9.

Assumptions and Adversary Model. We use ProVerif [194] to model the protocols and attackers. While MKA supports multiple configurations and optional features, including flexibility in how the CAK itself is established and how members are authenticated, relevant automotive MKA specifications and AUTOSAR [195], [196] assume a pre-shared CAK with a fixed key server. We make the same assumption. For CANsec, both the currently proposed MKA-based CKA and IKA assume a pre-shared CAK. We therefore model both with a pre-shared CAK. We assume a by-the-rules attacker following the Dolev–Yao model with two variations: an external attacker who is not a member of the CA, and an internal attacker who controls a compromised member ECU and has access to its CAK.

Verified Properties. We verify the following properties against both attackers:

- $\mathcal{P}_a$: An attacker cannot cause an honest CA member to accept spoofed CA messages or SAK distribution as originating from another honest member.
- $\mathcal{P}_f$: An attacker cannot cause an honest CA member to accept replayed CA messages or SAK distribution as fresh.
- $\mathcal{P}_c$: An attacker cannot learn protected CA communication exchanged between other members.

Results. In MKA, CA members derive the same KEK and ICK from the shared CAK and can therefore obtain the SAKs distributed by the key server. The MKA-based CKA approach follows the same general structure. IKA establishes SAKs differently, but every CA member possessing the shared CAK can derive them using the CAK and KN. Thus, despite their key-management and key-establishment machinery, neither protocol introduces key or privilege separation among members of the same CA, or any other mechanisms that protect against internal attackers. Accordingly, $\mathcal{P}_a$, $\mathcal{P}_f$, and $\mathcal{P}_c$ hold against the external attacker but not the internal attacker. Neither automotive MACsec nor the current CANsec configurations therefore close $\mathcal{DR}1$.

Hypothetical Revocation Experiment. The results above identify the inability to exclude a compromised member and rekey the remaining group as a key limitation of the considered configurations. We therefore test whether adding this capability closes the internal-attacker gap. We model an idealized revocation mechanism that immediately removes the compromised member from the CA and securely distributes a new CAK and SAKs to the remaining members. After revocation, $\mathcal{P}_a$, $\mathcal{P}_f$, and $\mathcal{P}_c$ hold against both external and internal attackers, closing $\mathcal{DR}1$. Our goal is not to propose a revocation protocol, but to test whether this capability is sufficient to address the identified limitation and motivate future research into practical mechanisms that provide it.

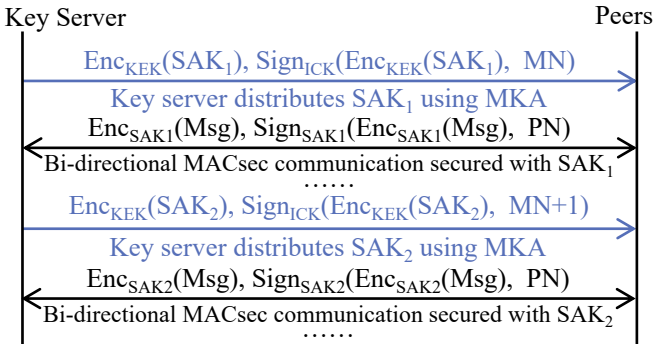

Fig. 8. MACsec and MKA.

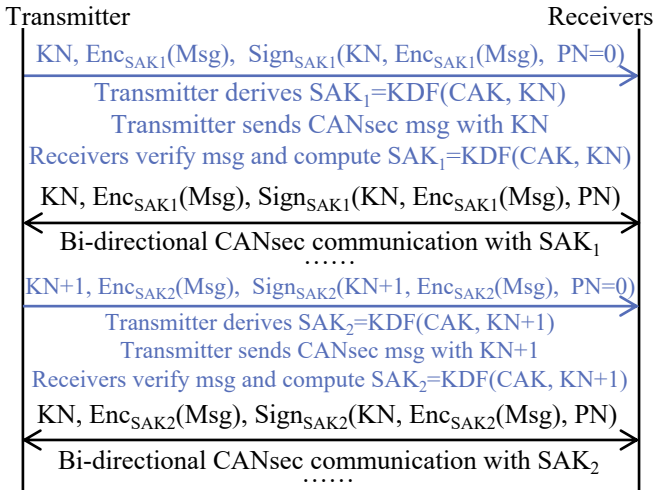

Fig. 9. CANsec and IKA.

Limitations. Our conclusions apply only to the configurations modeled above. MACsec supports additional authentication and key-management mechanisms that are not incorporated into the relevant automotive profiles and are outside our scope. Our results therefore should not be generalized to MACsec configurations using such mechanisms. CANsec and CKA remain under development, and our conclusions reflect the current proposal. They may differ for the eventual standardized design. The revocation experiment tests only whether ideal, immediate revocation is sufficient to close $\mathcal{DR}1$ and does not propose or evaluate a practical revocation protocol.