Securing the Future of Electric Vehicles: A Novel Approach Using MILS and Zero Trust Architecture

Jun Anzai [1] **Yoshiharu Imamoto** [1]

1) Panasonic Automotive Systems Co., Ltd., *Yokohama, Kanagawa, Japan*

E-mail: {anzai.jun, imamoto.yoshiharu}@jp.panasonic.com

ABSTRACT: The automotive industry is rapidly evolving due to technological advancements, particularly in vehicle architecture and CASE (Connected, Autonomous, Shared, Electric) technologies. Traditional Gateway (GW) architectures are transitioning to domain and zone architectures, facilitating the rise of Software-Defined Vehicles (SDVs). However, this integration increases security risks. This paper explores the application of Multiple Independent Levels of Security (MILS) and Zero Trust Architecture (ZTA) to vehicle systems. We propose a novel security approach to enhance vehicle system security and support sustainable evolution. This study aims to provide new perspectives for securing advanced automotive systems.

KEY WORDS: MILS, Zero Trust, Vehicle, SDV

1. INTRODUCTION

The automotive industry has undergone significant transformations in recent years, driven by rapid technological advancements. At the core of these changes are the evolution of vehicle architecture and CASE (Connected, Autonomous, Shared, Electric) technologies. Vehicle architecture is transitioning from traditional Gateway (GW) architectures to domain architectures, and further to zone architectures centered around high-performance computers (HPC)[7][8][13]. This integration enhances the collaboration between internal vehicle systems, enabling the realization of more advanced functions and contributing to software flexibility through Over The Air (OTA) updates.

The advancement of CASE technologies promotes the increase of connected services, integration with power grid systems, and the evolution of Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD). Vehicles are beginning to play roles beyond mere transportation, becoming part of the information and communication infrastructure. Furthermore, the concept of Software-Defined Vehicles (SDV) is spreading, intensifying the competition to continuously provide new value through software updates. The reason SDVs are gaining attention is that they define vehicle functions through software, allowing for flexible addition and modification of functions beyond hardware constraints, thereby significantly accelerating the pace of vehicle evolution.

In the paper[8], the architecture is examined in four quadrants divided into domain and zone types, and centralized and distributed processing systems, focusing on functional safety. However, security is not considered. As vehicle architecture integration progresses, security risks also increase. Various attacks and countermeasures have been proposed[13], but integration makes it easier for the impact of an attack on one subsystem to spread to other subsystems, resulting in numerous attack surfaces and assets to protect. In electric vehicles (EVs), compared to traditional internal combustion engine vehicles, there are fewer parts, and the integration of electronic control units (ECUs) is also progressing. In addition to individual security measures against various attacks, it is necessary to consider an appropriate security architecture. Traditionally, the widely adopted GW architecture[7][13] (Fig. 1) assumes that attackers may infiltrate the vehicle system through attack surfaces like the Telematics Control Unit (TCU) that connects to communication networks. Even if the ECU group located in the lower left, including the TCU, is infiltrated, the Gateway can defend against the intrusion, preventing access to the ECU group on the right. The right side contains ECUs with critical driving control functions such as "driving, turning, stopping," where the impact of an intrusion would be significant. Even if an intrusion occurs, the defense functions of the ECUs are expected to prevent it. An appropriate security architecture involves boundary defense and multi-layered defense. On the other hand, the domain architecture[7][8][13] (Fig. 2) consists of integrated ECUs grouped by roles, called Domain Controllers (DC). The paper[7] classifies DCs into four, while the paper[13] includes six, including the Gateway (Fig. 2). In the

domain architecture, attackers are assumed to infiltrate through the Connectivity DC, which groups communication functions like cellular networks/V2X, take over the Driver replacement DC, which integrates AD/ADAS functions, and ultimately aim to manipulate the Powertrain & vehicle dynamics DC, which controls driving. In domain architectures and further integrated zone architectures, the boundaries become ambiguous due to integration and increased attack surfaces, and hardware-level hierarchies decrease, making it inappropriate to rely solely on boundary defense and multi-layered defense. Therefore, new approaches are required to ensure the security of the entire vehicle system.

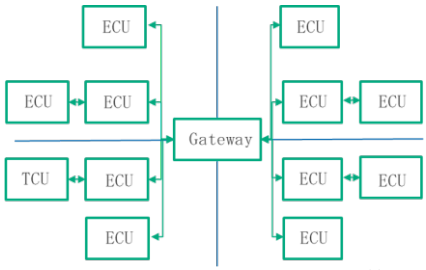

Fig.1 Gateway Architecture[7]

2. Previous Research and Challenges

As security architectures that replace or complement boundary defense and multi-layered defense, we introduce the classical yet continuously evolving Multiple Independent Levels of Security (MILS) architecture and the latest Zero Trust Architecture (ZTA).

2.1. MILS

The original MILS architecture was proposed in 1981[14], and it is a security concept for logically separating information and processes with different security levels. The main component of MILS is the Separation Kernel (SK), which separates and manages information and processes with different security levels to prevent interference. The security properties of SK - Non-bypassable, Evaluatable, Always invoked, Tamper proof (NEAT) - have been proposed[19]. Since then, it has been extended by many researchers[6]. For automotive applications, research mainly focuses on applying MILS to individual ECUs[3][12][18], but as pointed out in the paper[5], in zone architectures, distributed MILS[11] should be applied. In fact, a MILS architecture connecting partitions distributed across multiple ECUs via VPN has been proposed[9].

In this paper, we consider constructing MILS partitions on a domain architecture (Fig. 2). Here, similar to the assumptions of the D-MILS project[10] studying distributed MILS, partitions are configured as logical groups without considering hardware boundaries like DCs or Micro Processing Unit(MPU)s. By configuring logically, it is possible to follow frequent service additions and changes expected in SDVs. Many studies on MILS assume a single communication method, but further consideration is needed for vehicle systems where multiple communication methods coexist. For example, vehicles have different communication methods such as Controller Area Network (CAN), Ethernet, and Serial Peripheral Interface (SPI), but lack a unified management mechanism. Additionally, MILS operate based on pre-determined static policies, making it difficult to respond to dynamic context changes. Vehicle systems change context (e.g., stopping, charging, driving) based on driving conditions and sensor information, requiring a mechanism to dynamically change policies.

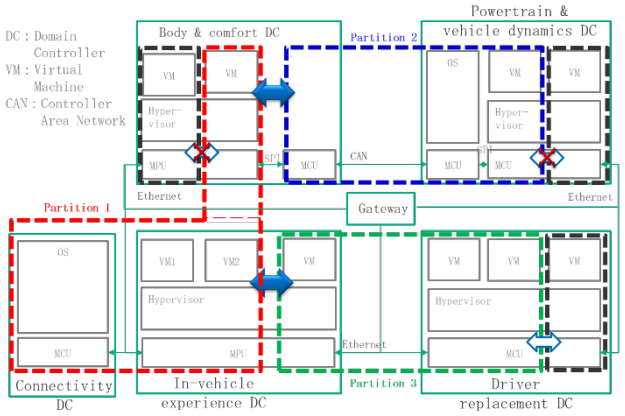

Fig. 2 Domain Architecture[13] and Partitions[10]

2.2. ZTA

ZTA is a security concept based on the principle of "never trust, always verify" for all resource requests. In this paper, we use the definitions and specifications from literature[15]. ZTA replaces traditional boundary-based security models by verifying all access, never trusting, and always confirming, thereby enhancing security through continuous verification, authentication, and authorization for users, devices, applications, and data. The main components of ZTA are the Policy Enforcement Point (PEP) and Policy Decision Point (PDP). PEP is the point where policies are enforced, receiving access requests and sending them to PDP. PDP dynamically decides access permissions based on policies. ZTA has been mainly adopted in cloud computing and enterprise networks, but its application to various fields, including embedded systems, is being considered[1][17]. However, research on applying ZTA to automotive systems is limited, and as far as the authors know, direct application to vehicles is only found in research on centralized ZTA for CAN-based systems[2], with several studies considering systems including V2X and cloud[20].

In vehicle control, real-time performance is crucial, but if PDP/PEP is centrally located in ZTA, ensuring real-time performance is challenging. Particularly, since each ECU in the

vehicle needs to collaborate in real-time, centralization may cause delays. Additionally, vehicle systems contain a mix of CPUs, ECUs, and DCs with varying performance levels, from smartphone-level MPUs to Micro Controller Unit (MCU)s without operating systems. These use different communication methods (CAN, Ethernet, SPI, etc.), making it difficult to apply a unified security policy.

3. Application of Security Architectures to Vehicle Systems

This chapter considers solutions to the challenges mentioned in Chapter 2.

3.1. Application of MILS to Vehicle Systems

In this paper, to address the coexistence of different communication methods like CAN, Ethernet, and SPI, we propose controlling communication through an intermediate area that converts multiple communication methods. This intermediate area performs protocol conversion for each communication method and applies a unified access control policy. The intermediate area is assumed to support two or more communication methods. For example, as shown in Fig. 2, the MCU in the Body & comfort DC, which connects to the CAN bus, also connects to the MPU for running applications via SPI communication, making it usable as an intermediate area. Additionally, in Fig. 2, the MPU in the Powertrain & vehicle dynamics DC runs applications with functional safety in a Virtual Machine (VM) separated by a Hypervisor, which realizes Ethernet communication and VM-to-VM communication, making it usable as an intermediate area.

3.2. Application of ZTA to Vehicle Systems

In vehicle systems, especially driving control needs to be performed in real-time. ZTA verifies all access through authentication and authorization, which may cause delays and compromise real-time performance.

As introduced in papers[15][17], one implementation strategy for ZTA is the use of micro-segmentation. Unlike traditional large-scale segmentation like DMZ (DeMilitarized Zone) in websites, micro-segmentation divides functions at a finer granularity within the cloud. By applying ZTA policies to each micro-segment on a vehicle, it is expected to reduce the processing load on PDP and minimize the impact on other micro-segments even if one micro-segment is compromised.

3.3. Application of MILS and ZTA to Vehicle Systems

MILS, by statically partitioning the system and enforcing strict access control based on these predetermined partitions, can enhance the security of vehicle systems. However, in SDVs, various services will be continuously updated or added. These services are realized through the coordination of multiple functions within the vehicle system, yet under MILS it is difficult to update access control policies flexibly in accordance with new services. Moreover, vehicle systems are frequently used by multiple individuals − for example, in family cases with several drivers or in car-sharing scenarios. This necessitates policy changes based on the occupant as well as adjustments according to the vehicle's location and operating conditions (e.g., temporary stops, engine off, charging, highway driving, etc.), which MILS is not well suited to handle in a flexible manner.

On the other hand, ZTA is capable of accommodating the flexible policy modifications required by vehicle systems. However, any compromise in the security resilience of a vehicle system may have serious implications for human safety; particularly, policies related to driving control must be maintained with utmost rigor. In terms of strict policy management, MILS is more appropriate.

4. Proposed Method

In this chapter, we propose a security architecture that combines the strict policy management of MILS with the flexible policy management of ZTA for vehicle systems, particularly SDVs, to achieve strict yet flexible policy management.

4.1. Security Architecture Design

4.1.1. Definition of Functions

First, we describe the functionality of each component of the architecture.

SK: This is the foundational component of the MILS architecture that enforces isolation between partitions with different security levels based on a static policy. By doing so, each partition operates independently and is protected from influences coming from other partitions.

- Isolation: Strictly controls communications between partitions to prevent data leakage and unauthorized access.
- Resource Management Based on Static Policy: Appropriately allocates resources (CPU, memory, I/O, etc.) to each partition.

PDP: The Policy Decision Point is the component responsible for making access control decisions. In MILS partitions, the PDP determines dynamic policies for inter-partition communication and resource access based on the results of context evaluation and dynamic policy.

- Policy Decision: It allows or denies communication requests between partitions or resource access requests based on a predefined policy.
- Context Evaluation: It assesses the context of the request (such as user roles, device status, timing of access, etc.) to apply an appropriate policy.

PEP: The Policy Enforcement Point is the component that enforces the policy determined by the PDP. In MILS partitions, the PEP applies the dynamic policy to communications and resource accesses within and between partitions, while also monitoring and logging sessions that have been permitted.

- Policy Enforcement: Based on the policy determined by the PDP, it allows or blocks intra- and inter-partition communications and resource accesses, disconnecting sessions as needed according to the monitoring results.
- Monitoring and Logging: It monitors and records logs for every access request, which is used for detecting and responding to security incidents.

4.1.2. Definition of Policies

In the proposed method, we extend the SK of MILS to integrate PDP/PEP of ZTA and set policies through the following steps. Fig. 3 shows the image of the integration.

Step 1: Definition of Policies

- Define policies in MILS and express part of them in a format understandable by PDP. These policies clarify which partitions or entities within partitions can access which resources and which communications between partitions are permitted. In other words, define policies for access to all resources within partitions. These policies are based on various factors such as occupants, vehicle system state, and security level of the accessed resource. The policies held by SK are classified into the following three:
  - Static Partition Separation Policies
  - Static Access Control Policies (Non-modifiable)
  - Static Access Control Policies (Modifiable)

  PDP holds the following policies:
  - Dynamic Access Control Policies

  Note that the terms “static” and “dynamic” are used as follows:
  - Static: The policy is predetermined and, in principle, does not change. In addition, it does not consider dynamically changing contextual information at the time of authentication. We use the term “static” because in the proposed method policies are divided into those that must remain unalterable for security reasons and those that can be changed only when certain conditions are met.
  - Dynamic: The policy can be added to or modified even after shipment. Furthermore, contextual information is considered during authentication. In a vehicle system, it is necessary to update and maintain policies appropriately according to the various contexts (for example, while driving, during maintenance, while charging, or when parked), since the required policy differs depending on the context.

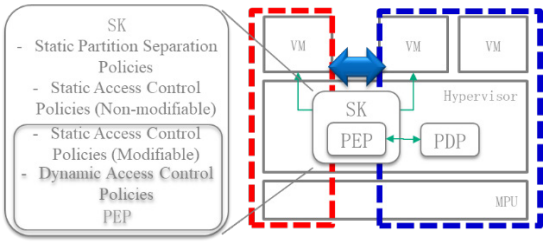

Fig.3 Integration of PEP into SK

Step 2: Extension of SK

- Extend the static access control policies (modifiable) in MILS to enforce dynamic policies of ZTA only when specific conditions are met. Here, collect real-time context information (driving conditions, speed, sensor information, etc.) within the vehicle and enforce dynamic policies based on it. For example, temporarily permit normally prohibited communication in emergencies. At this point, the SK communicates with the PDP to evaluate whether the access request complies with the policy. Next, the SK receives access requests from each component within the system as the PEP, communicates with the PDP to determine whether the requested operation is allowed, and then returns the decision to the corresponding PEP component to allow or deny the access request.

Step 3: Continuous Authentication and Authorization

- By integrating the PEP functions within SK and linking them with the PDP, the system continually evaluates access requests and determines access permissions based on policy. This includes verifying user authentication credentials, checking the security status of devices, and confirming that the security requirements of the resource to be accessed are met. Each time there is a change in the system context, SK re-evaluates the policy and ensures compliance. This approach maintains the overall security level of the system and provides the flexibility necessary to respond to dynamic threats.

4.1.3. Policy Use Cases

Using Fig. 2, three use cases (UC) are explained in which a dynamic policy is enforced on top of a mutable static policy.

UC1: Service to Check Charging Status on a Smartphone -

Assume that charging is performed at the Powertrain DC. Under the mutable static policy, communication from Partition 1 to Partition 2 is prohibited, while periodic transmission of battery status from Partition 2 to Partition 1 is allowed. The dynamic policy then permits a request to retrieve charging status information from Partition 1 to Partition 2 on the condition that the vehicle system’s state is connected to an EV charger and in a charging state.

UC2: Maintenance Service – Assume that maintenance is implemented at the Body & Comfort DC. Under the mutable static policy, communication from Partition 1 to Partition 3 is prohibited. The dynamic policy permits the sending of maintenance commands from Partition 1 to 3 provided that the vehicle system is connected to a maintenance tool (indicating a maintenance state) and is in a parked condition.

UC3: Intrusion Detection – When an intrusion is detected in Partition 1, the dynamic policy prohibits access from Partition 1 to any other partition. In this case, even if the conditions for UC1 or UC2 are met, their dynamic policies are not enforced; instead, the dynamic policy for UC3 is given priority. Moreover, requests for resources within Partition 1 are subjected to additional authentication on top of the usual authentication. For instance, the integrity of the process being authenticated may be verified.

4.2 Detailed Design

4.2.1. Response to Attack Detection

In the event that the PEP/PDP is attacked, we propose the following two countermeasures. If the PDP is not compromised, use enhanced mode; if it becomes compromised, switch to safe mode.

Safe Mode: Cease enforcement of dynamic policies and apply only the static policies provided by SK. In anticipation of a compromise of the PEP/PDP, backup PEP/PDP modules are prepared redundantly for rapid switching. This approach enables the system to quickly return to dynamic policy application after exiting safe mode.

Enhanced Mode: Predefine rules to strengthen the dynamic policy. For example, if unusual accesses are detected, access control is tightened to restrict certain operations. This allows the system to respond swiftly to attacks and minimize potential damage.

4.2.2. Ensuring Real-Time Performance

To reduce the processing load on the PDP, we propose the following three measures:

Priority Setting: Separate tasks that require real-time performance from other tasks by assigning them high priority. In addition to real-time tasks, security-critical tasks are given high priority, ensuring that they execute preferentially over other tasks. Specifically, for driving control, only the static isolation and access control provided by MILS are applied; dynamic access control by the PDP is not applied.

Caching: Add a cache to the PEP to store frequently used policies or evaluation results. When there is a cache hit, the evaluation result is quickly retrieved from the cache, shortening the evaluation time. It is also possible to pre-calculate and cache evaluation results for frequently used policies during system startup. To ensure the integrity of the cache, digital signatures or MACs are attached.

Distributed Processing: In a system with a single PDP, processing may become a bottleneck if all tasks are centralized. By distributing the PDP functions across multiple nodes (referred to as “edge PDPs”), the processing load is distributed. Note that there is a proposal[16] for distributing PDP functions with a focus on resilience.

4.2.3. Placement of PEP and PDP

The unified SK (with integrated PEP functionality) is placed in a location that achieves separation between partitions. As shown in Fig. 4, if partitions span across physical ECUs, an SK (with PEP) is placed on each corresponding physical ECU. Moreover, if multiple microcontrollers exist within an ECU, an SK (with PEP) is placed on each microcontroller.

On the other hand, we propose three placements for the PDP:

Basic Configuration: A single PDP is deployed on the vehicle system and is accessed by each PEP.

Distributed Configuration 1: As shown in Fig. 4, one Master PDP (M-PDP) is deployed on the vehicle system, while multiple Edge PDPs (E-PDPs) are deployed. In a zonal architecture, an E-PDP may be placed on each zonal ECU. Alternatively, without regard for the physical architecture, an E-PDP may be deployed per partition. Although increasing the number of E-PDPs can reduce the processing load, each E-PDP holds different information and may not share attack information. Although the information from the E-PDPs can be aggregated by the M-PDP and distributed periodically to update their data, the time lag in this update might result in a lower security level due to the absence of the latest attack information.

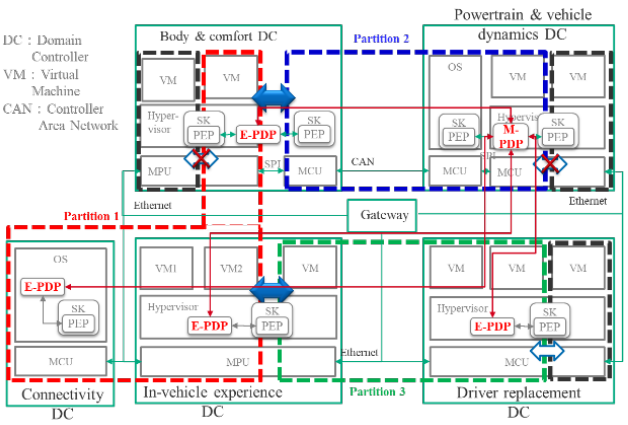

Figure 4 Distributed PEP and Distributed PDP

Distributed Configuration 2: E-PDPs are deployed in the same manner as in Distributed Configuration 1; however, no M-PDP is assumed. Instead, the E-PDPs periodically synchronize with one another to share information and form a virtual M-PDP. This approach ensures that even if one E-PDP is compromised (a risk in Distributed Configuration 1 with an M-PDP), the attack can be mitigated by consulting information from other synchronized E-PDPs. This approach enhances the overall resilience of the system.

5. EVALUATION

5.1 Characteristics for Comparison

In evaluating the proposed approach, we first define the characteristics that a security architecture for vehicle systems should satisfy. In this paper, it is assumed that the static partitioning and access control policies included in SK − as well as the functions that enforce them − can withstand an attack for a certain period. This "certain period" is defined as the duration during which it is possible to detect and respond to the attack from partitions other than the one under attack.

Security: In this paper, "security" for a vehicle system refers to the property that the system and its data are protected from unauthorized access, leakage, modification, or corruption. It is evaluated based on the following two elements.

- Minimization of Impact: This means that if an intrusion occurs in one area of the vehicle system, the degree of influence on other areas is very low. This concept includes MILS's "Damage Limitation" as well as ZTA's resistance to lateral movement. For example, in Fig. 2, the attacker's goal is to, after infiltrating Partition 1, intrude into Partition 3 and thereby manipulate Partition 2. Under this property, the effectiveness of preventing intrusion from Partition 1 to Partition 3 is measured.
- Attack Response Capability: This refers to the extent to which a security function can respond when under attack. "Response" here means the countermeasures that are purposefully executed to mitigate the effects of the attack. Generally, this includes processes such as disconnecting the communication that initiated the attack or deleting the compromised process. In access control methods such as MILS or ZTA, it refers to the ability to change the policy (e.g., prohibiting the communication that triggered the attack or blocking resource requests from the compromised process).

Real-time Performance: This characteristic refers to the system's ability to complete processing by a required deadline, which is particularly important in vehicle systems due to the need to interface with AD/ADAS and driving control functions.

Availability: This is the property that a system or service is accessible when needed. In this paper, it means that even if some of the security functions become compromised, other parts can either continue to ensure security or take over the security functions.

Flexibility: This property refers to the system's ability to adapt to changing requirements or environments. For vehicle systems, it signifies the ability to update security functions to protect newly added functions or services − for example, the ease with which partitions or policies can be modified. It also encompasses the ability to support multiple ECUs, different ECU configurations (such as those combining multiple microcontrollers), and various communication methods. For instance, supporting multiple communication protocols such as CAN, Ethernet, SPI, etc.

5.2 Introduction of Existing Methods for Comparison

The following four existing methods are introduced as comparisons to the proposed approach.

Vehicle MILS_1: This is the MILS approach targeting a single on-board hardware system as described in papers[3][12][18].

Vehicle MILS_2: This refers to the MILS approach targeting multiple on-board hardware systems as described in the paper[9]. In this case, VPN is assumed as the communication method between hardware. Note that the paper[5] mentions the necessity of D-MILS for vehicles. The D-MILS project[10] proposes connecting MILS hardware via an Ethernet specification − such as Time-Triggered Ethernet − that is capable of ensuring short latency and losslessness. As a result, applications to the smart grid[4] and air traffic control management systems[11] have been demonstrated. In this chapter, the method described in the paper[9] and D-MILS applied to vehicle systems are treated as equivalent.

Vehicle ZTA_1: This is a method for applying ZTA to vehicles as indicated in the paper[2]. It assumes a structure in which a centrally located PDP coexists with multiple ECUs connected via a CAN BUS.

Vehicle ZTA_2: Although not explicitly proposed as a method, by applying micro-segmentation to Vehicle ZTA_1[15][17] and combining this with distributing trust among multiple PDPs (referred to in the literature as "distributed PDP") to achieve resilience[16], it is expected that assigning a distributed PDP for

each micro-segment will enhance minimization of impact upon an intrusion, improve real-time performance, and increase availability during an attack. This method is also used as a reference for comparison.

5.3 Comparison between the Proposed Method and Existing Methods

Table 1 shows the comparison results, and each comparison item is explained below. Here, "++" means that the characteristic is fully satisfied, "+" means that the characteristic is partially satisfied, and "N/A" indicates that the characteristic is not satisfied.

Security1 (Minimization of Impact): Vehicle MILS_1 applies MILS only on a per-hardware basis; while it prevents intrusion between partitions on the same hardware, it cannot prevent intrusion into partitions on other hardware or across hardware. In Vehicle ZTA_1, although continuous authentication by the PDP can keep the impact to a certain extent even after an intrusion, it is difficult to prevent intrusions into other hardware or partitions.

The proposed method, Vehicle MILS_2, and Vehicle ZTA_2 are each capable of enforcing access control on partitions across different hardware as well as across hardware boundaries.

Security2 (Attack Response Capability): Both Vehicle MILS_1 and Vehicle MILS_2 are unable to respond to attacks. On the other hand, Vehicle ZTA_1 and Vehicle ZTA_2 are capable of appropriate authentication and access control by the PDP in response to attacks, yet since the PEP/PDP itself cannot respond if it is attacked.

The proposed method not only provides an enhanced mode to respond to attacks but also switches to a safe mode when the PEP/PDP is attacked, thereby ensuring that at least the driving control is minimally protected by MILS.

Real-time Performance: Both Vehicle MILS_1 and Vehicle MILS_2 support real-time performance guarantees through MILS. In contrast, Vehicle ZTA_1, where processing is concentrated on a single PDP, becomes a bottleneck for real-time performance. Vehicle ZTA_2, with distributed PDPs that help spread out the processing load.

The proposed method utilizes MILS alone for the parts related to driving control (thus supporting real-time control) while applying ZTA for other functions along with distributed PDPs.

Availability: In Vehicle MILS_1 and Vehicle MILS_2, even if a specific hardware or partition is compromised, the security functions on other MILS - enabled hardware or partitions function independently − but the compromised segment is rendered unusable. Vehicle ZTA_1, meanwhile, cannot guarantee anything if the PDP is compromised. For Vehicle ZTA_2, although the PEP is not distributed (one per partition), the PDP is distributed.

The proposed method, benefiting from MILS-based partition isolation and distributed PEP/PDP.

Flexibility: Vehicle MILS_1 is applicable only to a single hardware system. In contrast, Vehicle MILS_2, Vehicle ZTA_1, and Vehicle ZTA_2 can be applied across multiple hardware; however, their support for multiple communication methods is insufficient.

The proposed method, which can be applied across multiple hardware platforms as well as various communication methods.

In summary, while the proposed method shows favorable characteristics in all categories, it stands out particularly in terms of security (attack response capability), availability, and flexibility − attributes for which only the proposed method has received an "++" rating.

Table 1 Comparison of Proposed Method and Existing Methods

Characteristic	Proposed Method	Vehicle MILS_1 (3)(12)(18)	Vehicle MILS_2 (9)	Vehicle ZTA_1 (2)	Vehicle ZTA_2
Security1	++	+	++	+	++
Security2	++	N/A	N/A	+	+
Real-Time Performance	++	++	++	N/A	+
Availability	++	+	+	N/A	+
Flexibility	++	N/A	+	+	+

6. CONCLUSIONS

The proposed method demonstrates that by combining the strict policy management of MILS with the flexible policy management of ZTA, the security of vehicle systems can be improved. Specifically, we proposed a method that combines MILS-based partition isolation with dynamic access control provided by ZTA so as to support flexible policy changes while ensuring real-time

performance. Additionally, we presented specific implementation strategies, such as the distributed deployment of PDP/PEP, and the utilization of caching.

ACKNOWLEDGMENTS

I would like to express my gratitude to Mr. Ken Nicolson, who reviewed the paper and provided valuable comments.

REFERENCES

(1) M. A. Azad, S. Abdullah, J. Arshad, H. Lallie, and Y. H. Ahmed, "Verify and trust: A multidimensional survey of zero-trust security in the age of IoT," Internet of Things Volume 27, October 2024.

(2) J. Anderson, Q. Huang, L. Cheng, and H. Hu, "A Zero-Trust Architecture for Connected and Autonomous Vehicles," IEEE Internet Computing, Volume: 27, Issue: 5, Sept.-Oct. 2023.

(3) D. Adam, S.Tverdyshev, C. Rolfes, and T.Sandmann, "Two Architecture Approaches for MILS Systems in Mobility Domains (Automobile, Railway and Avionik)," International Workshop on MILS: Architecture and Assurance for Secure Systems (MILS 2015), 20.01.2015.

(4) D. Bytschkow, J. Quilbeuf, G. Igna, and H. Ruess, "Distributed MILS Architectural Approach for Secure Smart Grids," Second International Workshop, SmartGridSec 2014, February 26, 2014.

(5) A. G. Camek, C. Buckl, and A. Knoll, "Future cars: necessity for an adaptive and distributed multiple independent levels of security architecture," Info & Claims HiCoNS '13: Proceedings of the 2nd ACM international conference on High confidence networked systems, April 2013.

(6) R.J. DeLong, and E. Rudina, "MILS Architectural Approach Supporting Trustworthiness of the IIoT Solutions," An Industrial Internet Consortium Whitepaper, MARCH 10, 2021.

(7) M.Dibaei, X.Zheng, K.Jiang, R.Abbas, S.Liu, Y.Zhang, Y.Xiang, and S.Yu, "Attacks and defences on intelligent connected vehicles: a survey," Digital Communications and Networks, Volume 6, Issue 4, November 2020.

(8) A.Frigerio, B.Vermeulen, and K.G.W.Goossens, "Automotive Architecture Topologies: Analysis for Safety-Critical Autonomous Vehicle Applications," IEEE Access (Volume: 9), April 2021.

(9) D. Jedynak, "MULTIPLE INDEPENDENT LEVELS OF SECURITY (MILS) NETWORK REFERENCE ARCHITECTURE," 2015 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM, AUGUST 4-6, 2015.

(10) F. B. Kessler, and U.J. Fourier, "D1.3 Requirements for distributed MILS technology." Version 1.2(FinalPublic Distribution), August 2013.

(11) W.Kampichler, W. Steiner, and D. Eier, "DISTRIBUTED MILS: A NOVEL APPROACH TO ADVANCED ATM COMMUNICATION SERVICES," ICNS 2013, April 23–25, 2013.

(12) [12] M. Pitchford, "Applying MILS principles to design connected embedded devices supporting the cloud, multi-tenancy and App Stores," 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), Jan 2016.

(13) L. Reger, "The EE architecture for autonomous driving a domain-based approach," ATZelektronik Worldwide, vol. 12, pp. 1621, Dec. 2017.

(14) J.M. Rushby, "Design and Verification of Secure System," ACM SIGOPS Operating Systems Review, Volume 15, Issue 5, Pages 12 - 21, December 1981.

(15) S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," NIST SP 800-207, August 2020.

(16) B. Sengupta, and A. Lakshminarayanan, "DistriTrust: Distributed and Low-Latency Access Validation in Zero-Trust Architecture," Journal of Information Security and ApplicationsVolume 63, December 2021.

(17) N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "Zero Trust Architecture (ZTA): A Comprehensive Survey," IEEE Access (Volume: 10), 12 May 2022.

(18) S. Tverdyshev. "Security by Design: Introduction to MILS," embedded world 2017.

(19) G.M. Uchenick, and W.M. Vanfleet,"Multiple independent levels of safety and security: high assurance architecture for MSLS/MLS," MILCOM 2005 - 2005 IEEE Military Communications Conference, October 2005.

(20) M.Zayed, A.Anwar, Z.Rahman, Sk.S.Arefin, and R.Islam, "Owner Identity Verification in the Internet of Connected Vehicles: Zero Trust Based Solution," cryptoeprint:2022/1660.