Security-risk-mitigation Measures for Service Oriented Vehicle Diagnostics SOVD

Masaaki Miyashita
Cybersecurity Group, Connected Car Off-board Development and Operation Department of Nissan Motor Corporation
Kanagawa, Japan
e-mail: m-miyashita@mail.nissan.co.jp

Benchadi Djafer Yahia M
Cybersecurity Group, Connected Car Off-board Development and Operation Department of Nissan Motor Corporation
Kanagawa, Japan
e-mail: b-djafer@mail.nissan.co.jp

Hiroki Takakura
National Institute of Informatics
Tokyo, Japan
e-mail: takakura@nii.ac.jp

***Abstract*— A new vehicle diagnostic standard "Service Oriented Vehicle Diagnostics (SOVD)" is expected to be used for the next-generation vehicles known as Software Defined Vehicles (SDV). SOVD supports various vehicle maintenance demands, including remote diagnosis, by implementing web server function into a high-performance in-vehicle component. However, this architecture introduces additional security risks to SDV, as this web server functionality becomes a new cyberattack entry point into the vehicle. In this paper, we present several security-risk-mitigation measures for such systems, extending our previous work. Specifically, we propose multi-layered defense measures including physical and logical isolation (Zone Separation) of the web server software from security-critical software modules and in-vehicle HMI-based authorization for critical diagnostic privileges. We conclude that these additional security measures significantly reduce the feasibility of remote cyberattacks against SOVD-based remote diagnostic systems.**

Keywords-Automotive cybersecurity; Remote diagnosis; UDS; SOVD.

I. INTRODUCTION

This work is a follow-up to our prior work "Security-risk-mitigation Measures for Automotive Remote Diagnostic Systems", published in the proceedings of SECUREWARE2024 [1]. As technology advances, the electronic systems in automobiles are becoming more intricate. These systems consist of numerous components that are connected through in-vehicle communication networks. Diagnostic systems specifically designed for vehicles are required to pinpoint any malfunction. These systems usually require a diagnostic tool to be directly connected to a dedicated connector on the vehicle and must be operated at a garage.

With wireless communication systems increasingly used in vehicles, remote diagnosis systems have become more prevalent. These services enable an operator to read diagnostic trouble codes and data logs through wireless communication. This prompts the driver to bring his/her vehicle to a garage for repairs before the trouble becomes more severe. Diagnostic communications are used not only to read such data but also to write data to in-vehicle parts, such as firmware updates and initial settings of replacement parts.

Studies have indicated that cyberattacks targeting vehicles through diagnostic communications can result in significant damage. For example, it has been demonstrated that some diagnostic Controller Area Network (CAN) messages impacted major critical vehicle control systems, such as the engine, brake, and steering systems [2]. Car theft and privacy breaches are also potential risks of cyberattacks through diagnostic communication [3].

On the other hand, European Union vehicle type approval regulation EU 2018/858 [4] Annex X requires that "Manufacturers shall provide to independent operators unrestricted, standardised and non-discriminatory access to vehicle OBD information, diagnostic and other equipment, tools including the complete references, and available downloads, of the applicable software and vehicle repair and maintenance information. Information shall be presented in an easily accessible manner in the form of machine-readable and electronically processable datasets. Independent operators shall have access to the remote diagnosis services used by manufacturers and authorised dealers and repairers.", if a vehicle has the remote diagnostic system. This requirement makes designing measures against unauthorized access complex, because their network access routes and credentials for user authentication become various.

In our previous work [1], we mainly focused on risk mitigation for conventional remote diagnostic architectures based on UDS communication. This extended study introduces a new perspective by addressing the security challenges of the emerging Service-Oriented Vehicle Diagnostics (SOVD) framework. Building on this shift, we present security-risk-mitigation measures specifically adapted to the SOVD-based remote diagnostic systems. These systems involve reading diagnostic trouble data and remote firmware-update tasks that were previously only executed at service stations. Our measures aim to reduce the potential security risks associated with these systems.

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

The rest of the paper is structured as follows. In Section II, we discuss automotive diagnostic communication. Section III presents the current status and existing issues of remote diagnosis. In Section IV, we propose our security-risk-mitigation measures. In Section V, we show how to avoid constraints when implementing proposed measures in vehicle component. Section VI illustrates our prototype simulation for evaluation. Finally, we conclude our work in Section VII.

II. AUTOMOTIVE DIAGNOSTIC COMMUNICATION

The process of remote diagnosis involves the use of wireless communication between a vehicle and a diagnostic server located outside the vehicle. To diagnose the various components implemented in the vehicle, the in-vehicle wireless communication unit, which serves as the entry point to the vehicle, must communicate with other components through the in-vehicle communication network. To achieve this, it is most reasonable from a system-implementation standpoint to use the diagnostic communication protocol typically used for wired-connected diagnostic tools. While this protocol is effective for wired communication, there are security concerns when using it for wireless communication.

With this in mind, we examined the characteristics and issues of automotive diagnostic communications used in the in-vehicle network.

A. Overview of Diagnostic Communication

In 1991, the California Air Resources Board mandated the implementation of the On-Board Diagnostics (OBD) connector to standardize vehicle diagnostic communications. Today, the OBD2 connector is the industry standard interface and can use several communication protocols. CAN communication is prevalent in vehicle-embedded processors, and there is a shift towards faster diagnostic communication using Diagnostics over Internet Protocol (DoIP)-based communication with an Ethernet physical layer [5]. To address the need for faster communication and accommodate the increased complexity of automotive software, ISO14229-1 standardized the Unified Diagnostic Service (UDS) Protocol, which is now used as a standard communication protocol by many automotive companies. However, as software complexity increases, so do security concerns, as outlined in previous studies [6] and [7] on DoIP.

In 2022, ASAM (Association for Standardization of Automation and Measuring Systems) released a new vehicle diagnostic communication API (Application Programming Interface) “ASAM SOVD v1.0.0” [8] targeting the new generation vehicles with Software Defined Vehicle (SDV) architecture. This SOVD requires the vehicle architecture shown in Figure 1, because SDV requires High-Performance Computer (HPC) to have some Virtual ECUs as software components for easy upgradability of the vehicle functions. HPC is a key component of this architecture, because it hosts the SOVD server as a hub of diagnostic communication. It is one of big difference from UDS that SOVD supports the remote diagnosis as a native standard service. And SOVD also consider reusing old vehicle Electronic Control Units (ECU) with UDS protocol by CDA (Classic Diagnostic Adapter) as a communication translator between SOVD API and UDS. SOVD will be applied to many new generation vehicles with SDV architecture because it will be a new international standard ISO-17978 by the end of 2025.

B. Diagnostic Tool

Advancements in diagnostic-communication hardware and software have brought about changes in diagnostic tools used to identify failures in vehicles. Handheld terminals with basic Liquid Crystal Displays (LCDs) had been commonly used for diagnostic communication before the spread of CAN communication. However, with the increasing number of vehicles supporting diagnostic communication and the complexity of systems due to the introduction of IP communication, developing software for specialized hardware has become inefficient. Thus, it is now common to use a Personal Computer (PC) or tablet in Figure 2 as a diagnostic tool and connect it to an OBD dongle through USB, Bluetooth, wireless LAN, etc.

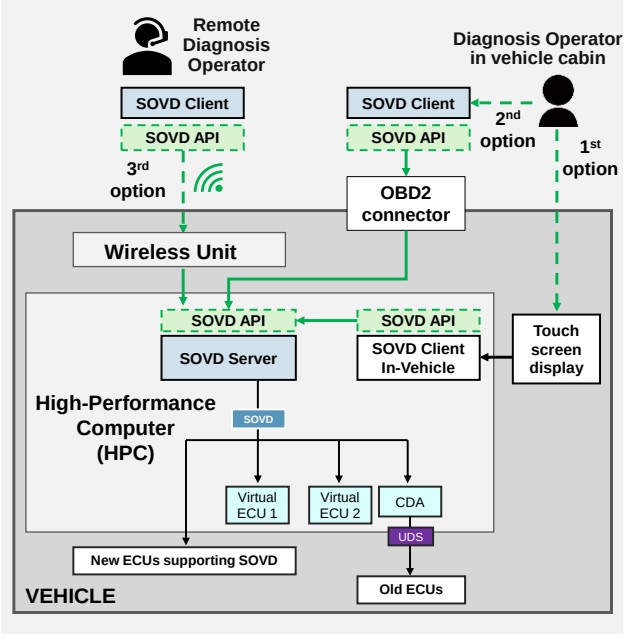

Figure 1. Example of SOVD vehicle architecture

This approach has the additional benefit of enabling developers of general diagnostic tools that support vehicles from multiple automobile companies to easily acquire diagnostic tool hardware. However, it also raises concerns

Figure 2. Diagnostic tools using PC/Tablet

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

that these devices, which are essentially PCs and tablets with network connectivity as standard equipment, could be used as gateways for attackers to intrude into vehicles. Since diagnostic communication protocols are standardized and diagnostic tools and software can be purchased inexpensively, attackers can find vulnerabilities through reverse engineering.

SOVD will also change the diagnostic tool. As Figure 1 shows, when a vehicle has a touch screen display in the vehicle cabin, SOVD can provide first option, “In-vehicle client” using the display to realize “Diagnostic tool-less” operation. A diagnosis operator can use access through the OBD2 connector as second option when the touch screen display is unavailable. When an operator requires access from a remote location, SOVD provides third option through the wireless interfaces. In this second and third options, any consumer devices (e.g., smartphones, tablets, personal computers) can access in-vehicle SOVD server by various web browsers, because the SOVD server uses REST (Representational State Transfer) API over HTTPS (Hypertext Transfer Protocol Secure) to communicate with the clients instead of the UDS which was used by old diagnostic tools. These options will give the diagnostic operator better flexibility than using the diagnostic tools, but requires stronger security measures, because implementing web server-client systems into the vehicle must bring new vulnerabilities as same as the information systems. In order to mitigate risks from old diagnostic tools, SOVD enforces vehicle-side authentication and authorization for all critical services. Conventional diagnostic tools in UDS system store ECU service information locally in the device and can be reverse engineered by an attacker. With SOVD, this risk is mitigated due to the diagnostic tool no longer contains vehicle-specific data or applications. All service information processes for ECU modification or software updates are handled by the in-vehicle HPC unit via the HTTPS server. This highly reduces the risk posed by traditional diagnostic tools to compromise ECUs.

C. *Security-critical Diagnostic Communication Services*

In UDS diagnostic communication, the functionalities offered by an ECU for using a diagnostic tool are referred to as "services". These services include reading and writing data to operate the ECU as well as diagnostic commands, such as fault code retrieval. The conversation surrounding automotive cybersecurity threats highlights the potential for attacks via the OBD connector by exploiting these services. Previous research [9] and [10] have demonstrated that the following UDS have been susceptible to exploitation.

- Input/Output Control Service: This service controls the input and output signals that are connected to the specified ECU from the diagnostic tool. Its primary function is to identify the failure point. For instance, if the wipers do not operate even after turning on the wiper switch, this service can be used to forcibly drive the wiper motor, and if the wipers start operating, it proves that the motor and its wiring have no problem. This approach helps in efficiently narrowing down the failure point. However, this service can lead to generating hazardous vehicle behavior that the driver did not intend.
- Write Data by Local ID Service: This service is designed for configuring the initial settings and adjusting the parameters of installed components. It can, for example, be used to write the dynamic radius value of a tire to the ECU to calibrate the speedometer or enable/disable optional parts. However, if this service is abused, users may experience adverse effects, such as inaccurate information display or suspension of certain functions.
- Reprogramming Service: This service is for rewriting ECU firmware installed in sold vehicles, usually to correct quality defects in the firmware. However, if this service is abused, it could result in various issues. For instance, the rewritten ECU may behave improperly or even spoof other ECUs, leading to more significant problems, such as sending malicious communication data to other ECUs. Therefore, it is crucial to use this service only for its intended purpose and avoid any abuse.

These services are locked by default as privileged operations within many UDS ECUs. To grant access to locked services, a process known as "security access (service ID27)" is typically used to verify the legitimacy of the user or diagnostic tool. New ECUs supporting SOVD will also have similar privileged services, and such services will be locked by SOVD server in HPC.

D. *Authentication by Service ID27 "Security Access”*

In diagnostic communication by using UDS, security access communication was generally executed using the following procedure (refer to Figure 3) with a pre-shared symmetric key K.

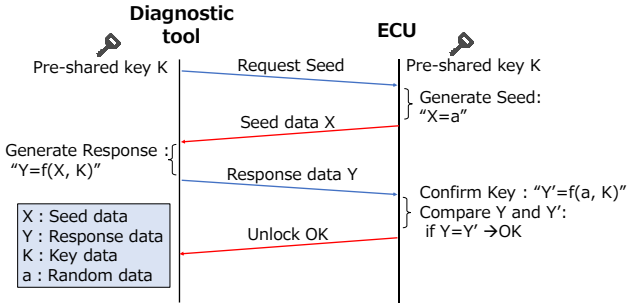

Figure 3. ECU unlock sequence by security access

1. The diagnostic tool to be authenticated sends a seed request (request seed) to the ECU to be unlocked.
2. Upon receiving the request, the ECU sends back seed data X, including random numbers, to the diagnostic tool to avoid the risk of replay attacks.
3. The diagnostic tool processes the obtained X using the key data K and computes the response data Y.
4. The diagnostic tool sends Y to the ECU. ECU calculates Y' from the K & X sent by ECU itself.

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

5. If Y' and Y are the same value, the authentication is successful, and the ECU unlocks the locked critical services.

If a symmetric key is used for authentication in security access executed by such procedures, an attacker may be able to obtain the key information through reverse analysis of the ECU or diagnostic tools. Therefore, the following solutions have been devised.

- To minimize the risk of reverse key analysis, it is essential to safeguard the private key in asymmetric key authentication. The private key should not be stored in the diagnostic tool. It instead should be kept in the Hardware Security Module (HSM), which is located on the authentication server or in a secure location with restricted access outside the tool. This requires the diagnostic tool to be connected to the authentication server with the HSM. To achieve this, infrastructure development and maintenance are necessary, such as installing a network environment at the garage and managing accounts that enable the diagnostic tool to log into the authentication server.
- Service ID27 does not provide security functions, such as user-privilege management or session key exchange with authentication, requiring each auto manufacturer to develop its own customizations. To remedy these issues, ISO 14229-1 has been updated, and a new UDS service, Authentication (Service ID 29), began in 2020.

E. Authentication by Service ID 29 "Authentication"

This new authentication service has the following advantages in terms of security compared with the previously used security access.

- Support for Public Key Infrastructure (PKI)-based authentication mechanisms.
- Support for session key exchange during authentication.
- User-privilege management support.

This service is expected to spread and be implemented into in-vehicle basic software, such as AUTOSAR (AUTomotive Open System ARchitecture). This will make it easier for vehicle manufacturers and component suppliers to implement higher security measures than ever before.

Some automotive ECUs, however, use processors with low processing power, such as 16-bit microprocessors. PKI-based authentication requires certificate parsing, hash calculation, and processing of asymmetric key cryptography, which cannot be afforded by such processors.

To introduce user-privilege management, it is necessary to properly construct and operate a system outside the vehicle that manages the privilege settings for each user and their expiration dates. For example, there is a need for special diagnostic communication during the vehicle-development phase and vehicle-production processes, and the introduction of Service ID 29 will not be effective unless account management for users and production facilities with such special privileges is properly implemented. Therefore, it is necessary to improve not only technical measures, such as the development of ECUs and privilege-management systems, but also the management and operation of the user management process at the same time.

F. Authentication of SOVD

SOVD solves the problem of low processing power ECUs by its centralized in-vehicle network architecture shown in Figure 1. SOVD server in HPC can authenticate the clients as a representative for all in-vehicle ECUs, because all diagnostic communication requests come in the SOVD server.

ASAM API specification [8] does not have a single standardized authentication method but has an informative specification using a Token base authentication and authorization.

III. CURRENT STATUS AND ISSUES OF REMOTE DIAGNOSIS

A. What is Remote Diagnostics?

Section II described wired diagnostic communication. Remote diagnosis refers to diagnostic communication using a wireless communication unit installed in the vehicle, enabling remote diagnosis from a location away from the vehicle. Figure 4 shows a typical configuration for remote diagnosis.

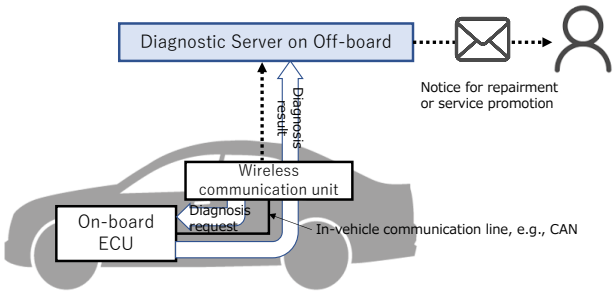

Figure 4. Example of remote diagnostic system

In remote diagnosis, the wireless communication unit in the vehicle requests the onboard ECU to self-diagnose if any failures occur. The onboard ECU sends back the diagnosis results, which the wireless communication unit forwards to the remote diagnosis server, enabling the diagnosis results to be obtained without entering the vehicle.

If a malfunction occurs, the diagnostic server notifies the user and urges them to repair or go to a garage, preventing the malfunction from becoming a serious problem.

While it is technically possible for the wireless communication unit to transmit requests, such as program rewriting and Input-Output (IO) control, these requests are designed for use under the control of a mechanic only when the vehicle is stopped for maintenance or repair. If operated remotely and unintentionally by the driver while the vehicle is running, they may cause safety-related problems.

In a previous study [11], security measures for remote diagnostic systems were proposed. These measures are based on the assumption that the wireless communication unit

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

(called the telematics module) is correctly installed in the vehicle and properly works. However, the vulnerability of the wireless communication unit can be exploited, making it an entry point for man-in-the-middle attacks through hijacking. This should be assumed as one of the major threats in recent automotive security risk analysis.

With current remote diagnostics, it is assumed that the wireless communication unit can be hijacked, thus the following risk mitigation measures were introduced as illustrated in Figure 5.

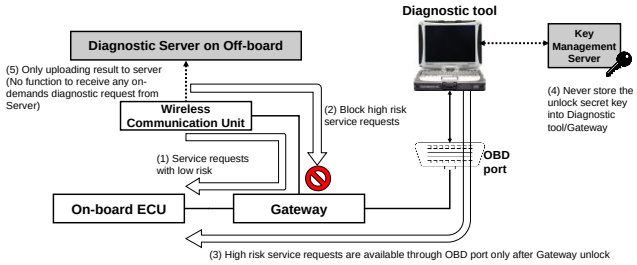

Figure 5. Example of conventional risk-mitigation measures

(1) The gateway is responsible for forwarding only low-risk service requests when the requests come from the Wireless Communication Unit, such as the reading of trouble codes and error log data. These available requests are registered in Gateway's static whitelist of authorized requests to prevent change it dynamically by any privilege escalation attack.
(2) If any high-risk service requests come from the Wireless Communication Unit, the gateway always blocks such requests because such requests are not in the whitelist.
(3) High-risk diagnostic service requests are available only by wired access through the OBD port after unlocking the Gateway's security protection. The in-vehicle network ports of Gateway for OBD port and Wireless Communication Units must be physically separated to identify the source of the service requests by the Gateway.
(4) The secret key required to unlock the Gateway protection are not stored in the diagnostic tool nor gateway to which the attacker can obtain physical access by purchasing them.
(5) The Wireless Communication Unit is not equipped with a function to receive arbitrary diagnostic requests on demands from an off-vehicle server but only uploads the diagnostic results. The Wireless Communication Unit should be able to transmit only predefined low-risk service requests to On-board ECU through Gateway, such as reading trouble codes.

B. Service Expansion Requirements for Remote Diagnosis

Contrary to the limitations imposed by the risk-mitigation measures described in Section III.A, the following use cases are required for remote diagnosis.

Use case 1: Remote use of critical commands (e.g., IO control services listed in Section II.C) required for pre-diagnosis to identify parts to bring to a repair place of a vehicle that is stopped on the road due to a malfunction.
Use case 2: Remote identification and handling of failure causes by senior mechanics (use case similar to telemedicine).
Use case 3: Remote diagnosis of whether a vehicle that has a trouble can be driven to a repair shop or whether it can be made drivable with simple road service assistance.
Use case 4: Understanding the status of a cyberattack (related to Section V.B.9).

C. Security Risks from Expansion of Remote Diagnostic Services

When responding to the need for service expansion as described above, the abuse of critical diagnostic services increases the risk that safety will not be maintained, and fatal incidents will occur.
Risk 1: Expanding the impact of incident occurrence: The impact of abusing critical diagnostic services becomes significant because such services can manipulate or illegally modify safety-related vehicle components, for example, the braking or steering system.
Risk 2: Failure to confirm the vehicle owner's consent and safe vehicle conditions: Conventionally, the owner's consent could be indirectly obtained by receiving the vehicle key to physically access the OBD connector inside the vehicle. The repair operator had to ensure that the vehicle was in a safe condition, such as by locking the wheels. By allowing work to be done remotely, the above measures cannot be used.
Risk 3: Risk of abusing remote operation authority: Conventionally, the OBD connector cannot be accessed unless the vehicle is physically in the hands of the mechanic, so there is no need to worry about workers to whom the owner has entrusted repairs in the past without the owner's permission. Remote operations do not have these restrictions, increasing the risk of insider attack by privilege holders.

To address these risks, the following countermeasures will be necessary:
Countermeasure against risk 1): To prevent the unlocking of critical commands through external communication only, a special in-vehicle operation for enabling remote diagnostics must be required as proof of the vehicle owner's consent.
Countermeasure against risk 2): In addition to electronically authenticating permission from the vehicle owner, the vehicle receiving the remote diagnostic command also checks the physical condition, indicating that the vehicle is not running but awaiting servicing as one of the conditions for conducting remote diagnosis.
Countermeasure against risk 3): When authenticating workers who conduct remote diagnosis, a mechanism to check whether the validity period of the work and the authority to carry out the work have been revoked is needed.

IV. Proposed Security-Risk-Mitigation Measures

An overview of the remote diagnostic system operation is shown in Figure 6. This system can execute remote diagnosis with the following procedure.

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A. Remote Operation Permission

The vehicle owner who wants to solve a problem with the vehicle or a mechanic who receives a repair request by the owner first conducts owner authentication in the vehicle. The following permission methods are possible.

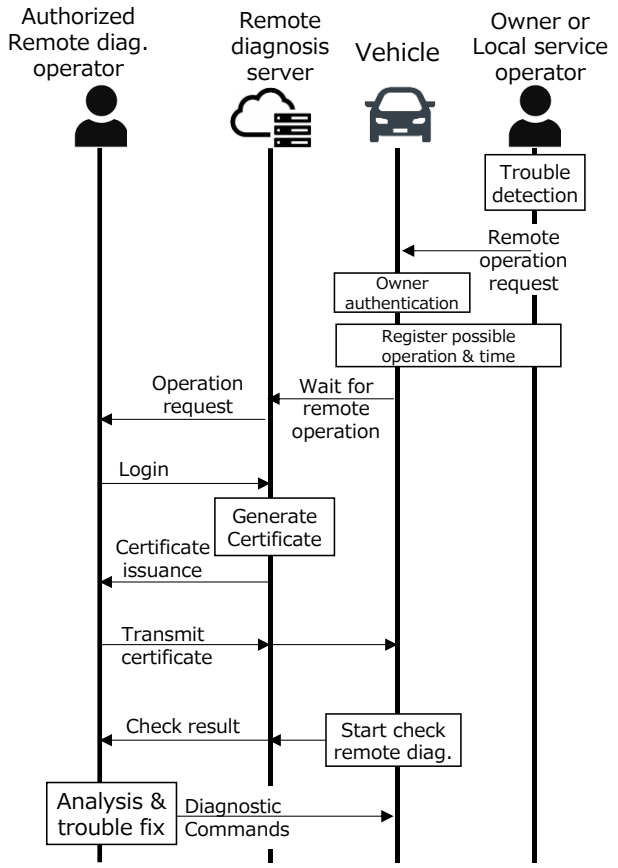

Figure 6. Overview of system operation

- The Human Machine Interface (HMI) in the vehicle (navigation-system screen, LCD of cluster meter, etc.) is used to authorize remote diagnosis. This can be done using a PIN or password preset by the vehicle owner to increase the reliability of the authentication.
- The presence of multiple intelligent keys in the vehicle is a condition for starting remote diagnosis permission. This is intended to detect differences from normal driving when only one key is present in the vehicle by the owner bringing a spare intelligent key into the vehicle.
- Pair the owner's smartphone with the vehicle and store the authentication information in the smartphone. The vehicle accepts remote diagnostics only for a certain period after successful Near Field Communication (NFC) authentication.

It is important to combine multiple conditions to increase the reliability of the remote diagnostic authorization described above.

B. Registration of Permitted Operations and Periods

Assuming that part of a vehicle component is malfunctioning, multiple input HMIs should be provided.

1) The owner's smartphone or operator's PC inputs the information and registers the operation information to be allowed to the remote diagnosis server and its validity period.
2) Input the information on an HMI in the vehicle and register the operation information to be allowed to the remote diagnosis server via the vehicle's wireless communication unit.

The user can select which operations to allow by using HMI of vehicle infotainment system or Web site of Remote diagnosis server, for example, reprogramming firmware or resetting the ECU.

C. Requesting Analysis via the Diagnosis Server

The remote diagnosis server notifies the target vehicle that the permitted operations and validity period of the work have been registered. At this time, the vehicle confirms that "permission for remote operation" has been granted in advance and that the vehicle is in a safe maintenance state (e.g., the vehicle is stopped, and the engine hood latch is open), and notifies the remote diagnosis server that it is "waiting for remote diagnosis".

The notification data from the vehicle can be supplemented with the vehicle's location information obtained from GPS, etc., and a request can be made to the diagnosis server to limit the locations where remote diagnosis is permitted to the area around the current location. Upon receiving this notification, the remote diagnosis server sends a failure-analysis request to an appropriate operator from among the "authorized remote diagnosis holders" registered in advance.

It is also effective to include a one-time password in the failure-analysis request to increase the reliability of the certificate-issuance process in the next step.

D. Generating and Issuing Certificate of Remote Diagnostic Operations

When an authority holder receives the notification, they log into the remote diagnosis server and request the issuance of a working certificate. To enhance security, it is recommended to require the entry of a one-time password, which is sent only to the authority holder when they receive the notification of the analysis request, as a condition for issuing the certificate.

The issuance of this certificate is also sent to an HMI of the vehicle and the registered smartphone of the vehicle owner. If this notification indicates that a remote diagnostic request was not intended by the driver or vehicle owner in the vehicle, the "waiting for remote diagnosis" status of the vehicle can be canceled, or an instruction can be sent to the remote diagnosis server to stop remote operation for the vehicle in question as a risk-mitigation measure.

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

The remote diagnosis server issues a certificate to the authority holder as a token that records the expiration date and permitted operating privileges.

E. Access to Vehicles from Remote-diagnostic-authority Holders

The authority holder responsible for remote diagnosis sends a token to the target vehicle. The vehicle checks the token's signature using the remote diagnosis server's pre-shared public key, and if the token is issued by the legitimate remote diagnosis server and is still valid, the vehicle unlocks the remote diagnosis communication and authorized operation rights recorded on the token. The expiration date on the token prevents unauthorized access after the work is completed, which is not intended by the owner.

V. Avoiding Constraints When Implementing Proposed Measures in Vehicle Component

A. Implementation Constraints to Consider

The following are constraints in implementing the proposed measures in a vehicle.

1. Automobiles are equipped with dozens of ECUs that execute diagnostic communications, and changing all these ECUs to components that implement security measures for remote diagnostics would require large-scale development and take too much time to implement.
2. The resources required to adopt enhanced authentication algorithms, user rights management and expiry date management cannot be implemented in components with resource-constrained processors, such as 16-bit microcontrollers, which limits their applicability.
3. Direct end-to-end communication between the off-vehicle server, which is the connection source for remote diagnosis, and the ECU to be diagnosed, creates a pathway for a direct attack on the ECU inside the vehicle from the off-vehicle server if a vulnerability exists in the ECU communication software, so a workaround is necessary.
4. Introducing SOVD architecture will be able to solve the constraints from 1 to 3 above, but it will bring other security risks, especially new risk caused by in-vehicle HTTPS server, because it makes a new attack surface having an open port to the internet.
5. Since SOVD uses REST API base communication, popular user authentication protocols (e.g., OpenID Connect [12], OAuth2.0 [13]) for web services would be preferable of the remote operator authentication. However, there is no available authentication service provider covering global vehicle markets to prove that the remote access requester is not a cyber attacker, but a skilled vehicle diagnostic operator, because proving it requires identity verification to check the requester's car maintenance experiences. Most of the vehicle manufactures want to avoid localizing the authentication system for vehicle development efficiency. Therefore, minimizing diversity of the remote operator authentication is an important demand of the remote diagnosis.

B. Our measures for UDS Generation to avoid Constraints

We devised our security-risk-mitigation measures shown in Figure 7 to avoid the constraints described in Section V.A.

To reduce the security risk of remote diagnosis, these measures have the following features that the conventional measures shown in Figure 5 do not have.

Measure 1: The in-vehicle gateway is used as the master ECU to manage the remote diagnosis control.

Measure2: The master ECU has a zone for communication with the external server via a wireless communication unit (Zone 1) and another zone for in-vehicle communication (Zone 2), which verifies certificate data for remote diagnosis and sends and receives diagnosis commands to and from multiple ECUs in the vehicle. Zones 1 and 2 are separated by hardware or software, such as a hypervisor, to prevent direct attacks from outside the vehicle to Zone 2, which executes in-vehicle communication processing.

Measure 3: Zone 1 of the master ECU communicates with the remote diagnosis server using Transport Layer Security (TLS) to prevent the in-vehicle wireless communication unit from eavesdropping on and falsifying communication data

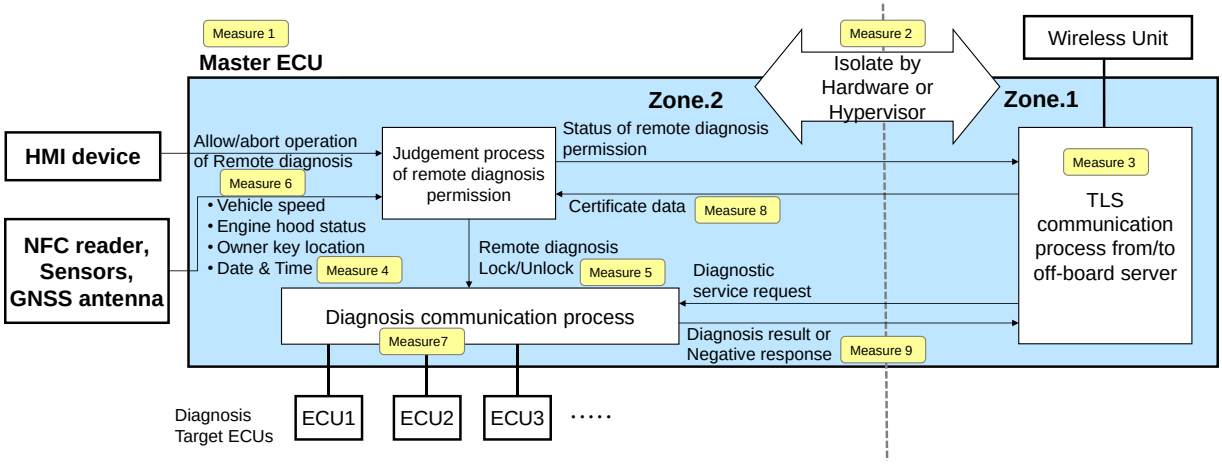

Figure 7. Implementation example for UDS generation using master ECU

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

between the master ECU and remote diagnostic server (a countermeasure against man-in-the-middle attacks).

Measure 4: To check the expiration date & time of the public key certificate for TLS, the master ECU must manage the absolute date & time using not only Global Navigation Satellite System (GNSS) data, but also trustable in-vehicle timer counter, because GNSS signals may get a replay attack. For example, the master ECU can detect the replayed GNSS signal when a newly received GNSS signal shows older time than the elapsed time of in-vehicle timer counter value or received signals in the past. Even if the timer counter's accuracy is low, e.g., a few seconds per month, it can still detect invalid GNSS signals when the difference between the result of adding the counter's elapsed time to the date and time of the last received GNSS signal and the date and time of the newly received GNSS signal exceeds the tolerance range.

Measure 5: The master ECU boots with the remote diagnostics as locked status by default. In the locked status, "Diagnosis communication process" in the master ECU rejects all diagnostic service requests coming from TLS communication process to prevent receiving any unexpected remote requests. Only when remote diagnosis is unlocked, the "Diagnostic communication process" in Zone 2 executes diagnostic communication in response to a remote-diagnostic-service request from Zone 1.

Measure 6: If the master ECU receives the result of the remote-diagnosis permission correctly executed with an HMI in the vehicle and the "remote diagnosis permission condition" is satisfied within a certain period after that, the master ECU unlocks the remote diagnosis process and enters the "waiting for remote diagnosis" state. The "remote-diagnosis-permission condition" is, for example, all the following conditions are satisfied.

(1) Successful verification of certificate received from Zone 1.
(2) The HMI executes remote diagnostic permission in the vehicle and is not canceled.
(3) No timeout has occurred since the operation in (2).
(4) The vehicle must be stopped.
(5) Signals indicating that the vehicle is in a service condition (e.g., engine hood is open) are detected.

Measure 7: The target ECU for remote diagnosis connected to the master ECU operates by receiving diagnostic commands from the "Diagnostic communication process" implemented in Zone 2. The master ECU executes the verification process of the certificate data and permission by the HMI, which are necessary as security measures of remote diagnosis, thus avoiding software and hardware changes in the target ECU.

Measure 8: If the verification of certificate data fails more than once, the time until accepting the next verification is extended.

Measure 9: If a diagnostic-service request that is not authorized by the certificate is received, the diagnostic communication process returns a negative response. This history is stored in remote diagnosis sever. The request commands thus rejected are signed and included in the negative-response history data to prevent repudiation by the authorized remote diagnosis operator.

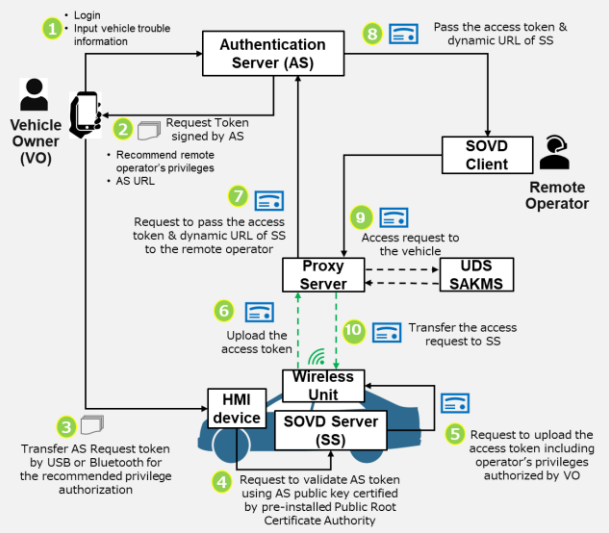

Figure 8. Example of remote SOVD sequence

In our previous paper [1], we inspected feasibility of implementing zone separation measures from a processing performance viewpoint. We conducted the experiment on Renesas R-carS4N-8A processor, and we confirmed that the proxy processing required for separating the zones could handle 96 Mbps of real-time video transfer with very low latency (1.675 ms), and we found no performance problem.

C. *Proposed SOVD Sequence to Address Constraints*

Figure 8 shows an example case of our proposal sequence to solve the SOVD constraints 4 & 5 described in Section V.A.

The SOVD generation will have the following sequence steps:

Step 1: A vehicle Owner (VO) can subscribe any remote diagnosis services (e.g., provided by the vehicle manufacturer, by a local car maintenance company etc.) and have its access account for remote service request. When VO wants to request for the remote diagnosis, the owner logins to Authentication Server (AS) and inputs the vehicle trouble information.

Step 2: Based on VO's input, AS generates a request token including a set of recommended remote operator's privileges. VO downloads this AS request token into VO's smartphone. If this request token is standardized among various remote diagnosis service providers and signed by PKI based certificate authority chain, the vehicle can manage the diversity of the service providers.

Step 3: VO transfers the downloaded AS request token to HMI device in the vehicle. HMI device extracts the recommended privileges and shows them in the touch screen display of HMI device for VO's approval. VO can accept them all or change them to a minimum set of privileges.

Step 4: After the approval by VO in step 3, HMI device sends the privileges authorized by VO to SOVD Sever (SS)

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

and requests SS to validate signature of AS request token. SS extracts the certificate chain information to get a public key of AS for the token signature validation.

Step 5: If signature of AS token is valid, SS generates an access token and requests Wireless Unit to upload it to AS by using AS URL in AS request token.

Step 6: The wireless unit passes the access token to the Proxy Server. This proxy server can be a bridge to UDS Security Access Key Management Server (SAKMS) of the vehicle manufacturer when an old UDS ECU sends a challenge to unlock its critical diagnostic operation. This bridge function can avoid connecting the remote client to the vehicle manufacturer's UDS SAKMS directly to get SA unlock response.

Step 7: The Proxy Server requests AS to send the access token to the remote operator. This Proxy Server hides the internet address of Wireless Unit by generating a random dynamic URL (Uniform Resource Locator) to prevent unexpected direct access to the vehicle from the public network. This dynamic URL is also shared with AS to inform it to the remote operator as a virtual URL of SS.

Step 8: AS passes the access token and dynamic URL to the remote operator.

Step 9: The remote operator starts accessing to the vehicle using the shared token & dynamic URL.

Step 10: The Proxy Server transfers the remote operator's access request to a target Wireless Unit specified by the dynamic URL.

As the Proxy Server hides the vehicle's access address from the public network, it can be the first firewall against the cyber-attack risk caused by constraint 4 in Section V.A.

Similarly, using PKI for the public key certificate chain at step 4 of this sequence enables covering various Authentication Servers in the global market, so it helps to solve constraint 5 in Section V.A.

An additional security advantage of this sequence is VO's approval by HMI device in the vehicle. This approval process requires some physical access actions in the vehicle cabin. This point can be a strong proof of VO's authorization.

Even though we implement security-risk-mitigation measures mentioned above, in-vehicle component also should have a similar zone separation as same as UDS generation in Figure 7 considering "Defense in depth" principle.

Figure 9 shows an example of the zone separation implementation for SOVD generation. In this example, "Public SOVD Server" in Zone-1 should provide HTTPS communication from/to the outside of vehicle through the Wireless Unit to mitigate the risk caused by constraint 4 in Section V.A. This "Public SOVD Server" has the similar functions of "Data communication process from/to off-board server".

Zone-2 hosts two new functions "Authorization Server" and "SOVD Manager" to match the SOVD software architecture.

"Authorization Server" has three token processes, the Request Token validation, the Access Token generation & check. The Request Token is authorized and validated by the vehicle owner's operation on HMI device and PKI Root CA Public Keys. If this authorization and validation are OK, the Access Token is generated using HPC's Private Key. When a remote operator starts access to the vehicle by sending its access token, "Authentication Server" also checks the access token sent by the remote operator. HPC must have a Secure Storage to protect the integrity of public keys and confidentiality of its private key.

"SOVD manager" has similar functions of "Diagnosis communication process" in Figure 7. It can lock or unlock the remote diagnostic communications using inputs from "Authentication Server" and "NFC reader, Sensors". "SOVD manager" switches the remote diagnosis communication path to the Virtual ECUs or old physical ECUs using UDS communication.

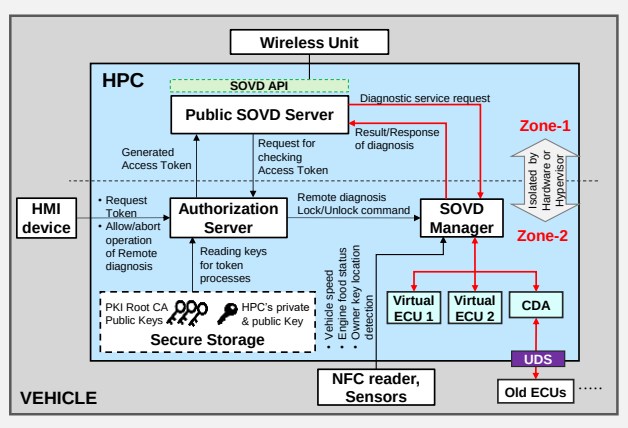

Figure 9. Implementation example for SOVD generation

VI. FUNCTIONAL SECURITY EVALUATION

Figure 10 illustrates the architecture of Proof-of-Concept (PoC) simulation environment based on our SOVD sequence described in Section V.C. In this evaluation, we focus specifically on the components highlighted in the figure, namely the Remote SOVD Client, the in-vehicle SOVD Public Server hosted on the HPC, and the Vehicle HMI. As these three elements represent the primary attack surface where authentication and authorization decisions occur.

Based on the simulation environment, we defined three functional test cases. The objective of evaluation is to verify the in-vehicle authorization concept functions correctly under realistic conditions. The evaluation focuses on two fundamental requirements:

Authentication Integrity: The SOVD Server must reject unauthenticated requests or invalid tokens, accepting only properly signed and valid credentials.

Authorization (scope enforcement): Access to diagnostic endpoints is determined by the operator's assigned privileges as reflected in the access token payload.

The following subsections detail the results of these functional test cases.

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A. Test Case 1: Unauthorized Access Token Blocking

In this test case, we verify that SOVD Server enforces token-based authentication correctly. We tested three scenarios:

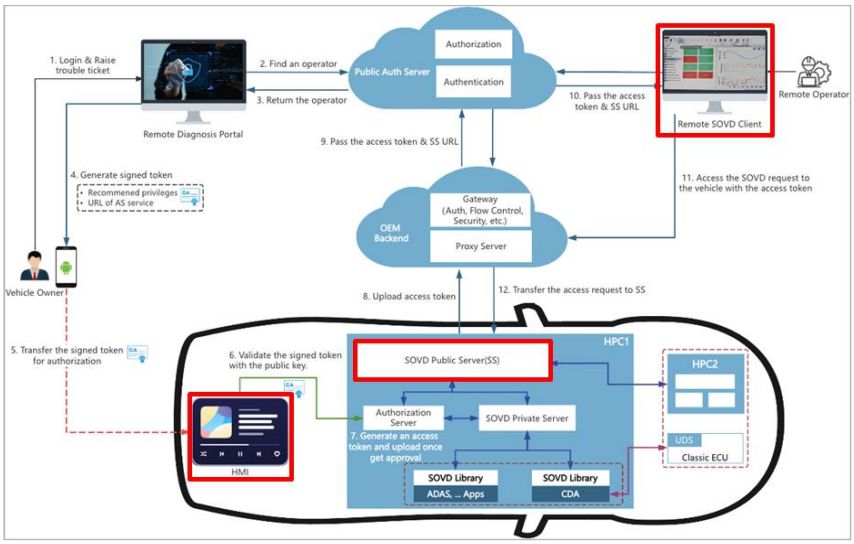

Figure 10. System Architecture of the SOVD Prototype Environment

1) Missing Authorization Header: A `GET` request was sent by the operator to the `/sovd/v1/Component` endpoint with an empty HTTP Authorization header. As shown in Figure 11, SOVD Server rejected the request with a "`401 Unauthorized`" response, which confirms that the Server does not allow unauthenticated operators to initiate any diagnostic query.
2) Invalid Token Signature: the same endpoint was accessed using a JWT token with an invalid signature. The Server returned "`401 Unauthorized`" as shown in Figure 12.
3) Expired Token: We attempted to access the API using a token that was structurally correct but possessed an expired timestamp. As Figure 13 shows, the Server responded with "`401 Unauthorized`" and identified the validation failure with a message "`claim timestamp check failed`".
4) Valid Token: Finally, a valid and correctly signed token was used. The Server responded with "`200 OK`", returning the structured component list defined by SOVD as shown in Figure 14.

B. Test Case 2: SOVD API Scope Enforcement

While Test Case 1 validates authentication integrity, Test Case 2 evaluates authorization by verifying whether protected SOVD API endpoints enforce the access permissions encoded in the JWT token. We define two operator roles: a "Viewer" role restricted to read-only operations, and a "Developer" role with full diagnostic permissions. Figures 15 and 16 show the permission configurations assigned to these roles in the simulation environment. To illustrate how these permissions are encoded in the issued credentials, we show a JWT token generated for a "Developer" operator in Figure 17. The token payload shows full access, including `GET`, `POST`, `PUT`, and `DELETE` operations, while the denyPermissions field is empty.

We then executed a functional test using the "Viewer" role. As shown in Figure 18, a "Viewer" operator successfully accessed fault information using `GET /sovd/v1/Components/adas-module/faults/C1456`, the operation succeeded with a "`200 OK`" response from the Server and returned the expected diagnostic data. However, when the same operator attempted to delete the fault code with a `DELETE` request, the Server correctly blocked the operation. Figure 19 shows the resulting "`403 Forbidden error`" with the message: "`Role 'Viewer' does not have permission to DELETE`". SOVD Server correctly interprets the permissions embedded in JWT token, and operations requiring elevated privileges (e.g., DELETE fault codes) are blocked for restricted roles.

C. Test Case 3: HMI UI Approval by VO

Finally, in Test Case 3 we validate the physical authorization step, to ensure that remote access cannot be established without explicit, in-vehicle approval by the Vehicle Owner (VO) through the in-vehicle HMI. Figure 20 shows the initial UI screenshot of HMI prompt displaying the list of permissions contained in the owner-provided certificate. Through the simulator, we show the requested privileges to VO to be selected (e.g., Body Control Module, HVAC, Power Line Communication). In this test scenario, VO selected only the "Body Control Module" permission and leave the other privileges unchecked. After approval, the HMI goes to confirmation screen shown in Figure 21,

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

indicating that “Remote diagnosis is in progress” with an option to “Quit Remote Diagnosis”. We confirm that the system enforces vehicle-side user approval before allowing any remote diagnostic activity.

D. Discussion

While the functional evaluations confirm that authentication, authorization, and in-vehicle approval mechanisms work as intended, we note that the timer counter's accuracy was not experimentally verified. However, our current vehicles already have decoded GNSS date and time information as in-vehicle CAN signals (e.g., via wireless communication unit). Therefore, even if the timer counter's accuracy is relatively low or GNSS time drift occurs, the system can still detect invalid GNSS signals when the difference exceeds a reasonable tolerance (e.g., several seconds). In practice, an attack that manipulates GNSS time by only a few seconds is highly unlikely, as such a minimal shift would not provide a meaningful advantage to an attacker. Consequently, the proposed detection approach remains effective even with coarse timer accuracy. Nevertheless, the evaluation does not yet quantify performance overhead or resilience against advanced attack scenarios such as DoS or token forgery. Future work should include large-scale stress tests, latency and resource profiling, and usability studies for HMI-based approval to validate feasibility in production environments.

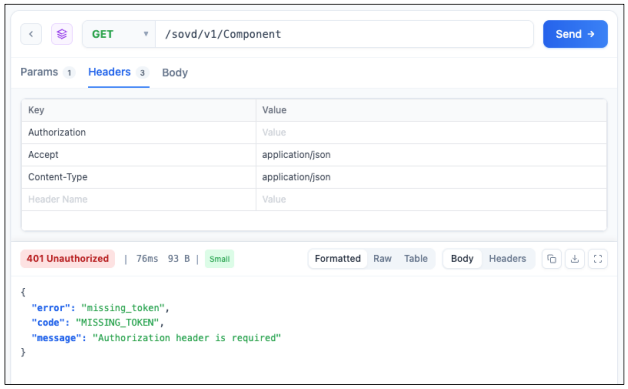

Figure 11. Response of an Empty Authorization Header

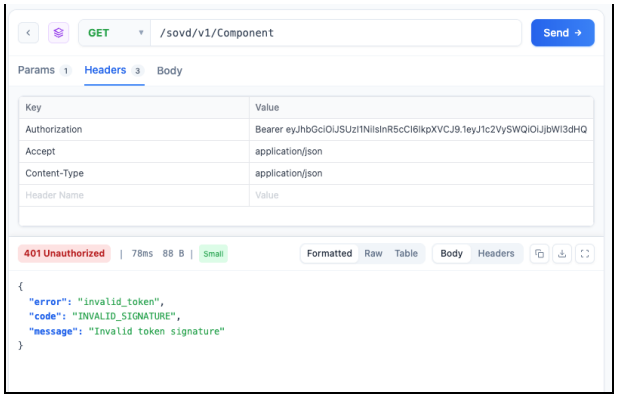

Figure 12. Response of Invalid Token Signature

Figure 13. Response of Expired JWT Token

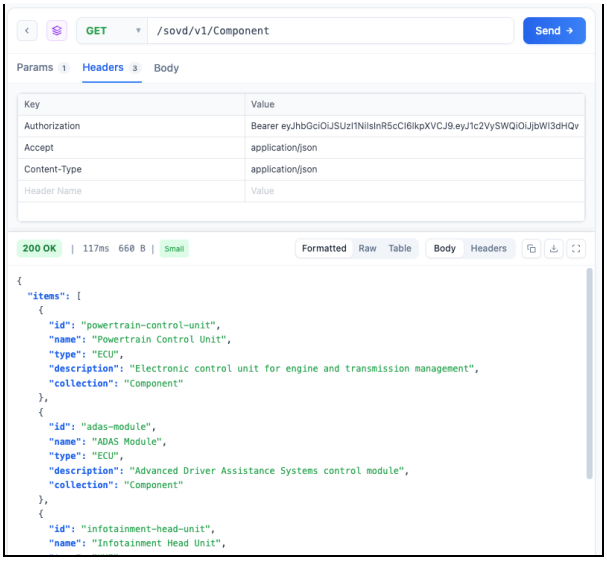

Figure 14. Response Success 200 OK with Component List

Figure 15. Viewer Role Permission Page

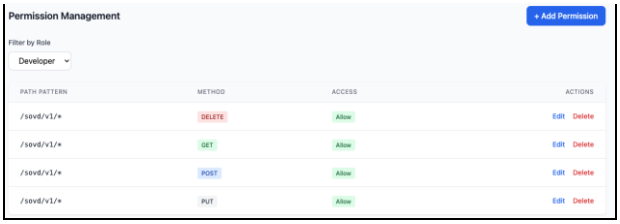

Figure 16. Developer Role Permission Page

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

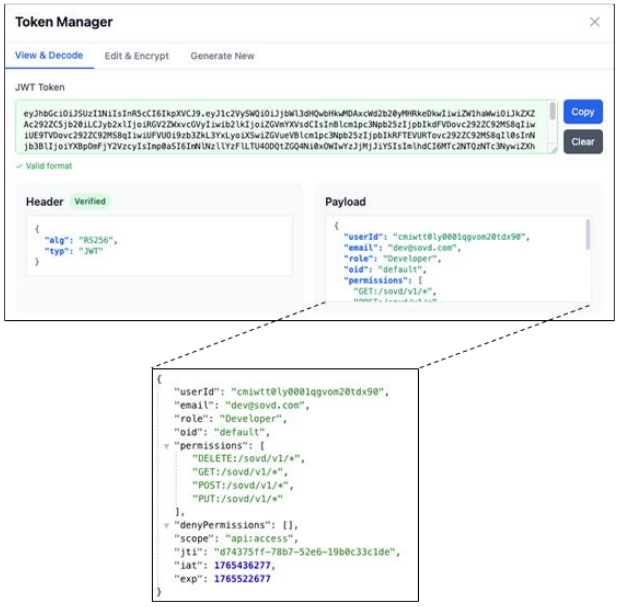

Figure 17. JWT Token Payload for Developer Role

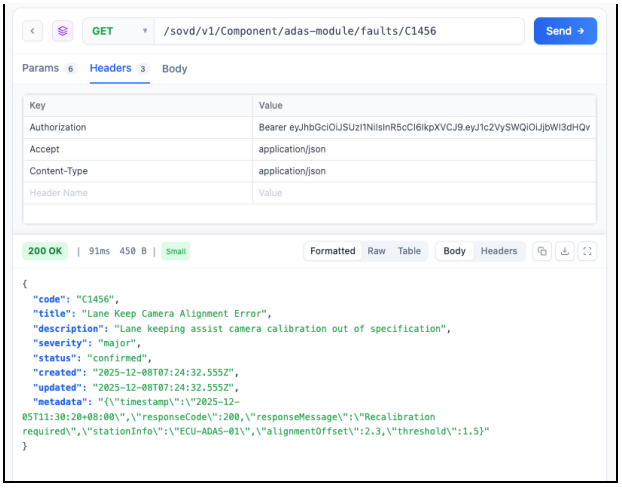

Figure 18. Viewer Operation “Read the fault info successfully”

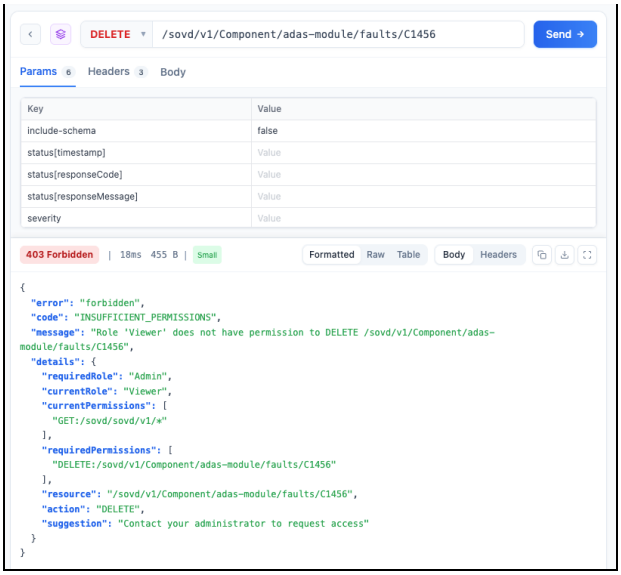

Figure 19. Viewer Operation “Failed to delete the fault”

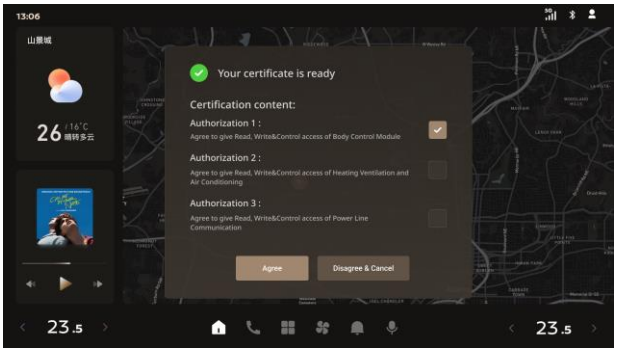

Figure 20. HMI Simulator UI for Selecting Diagnostic Privileges

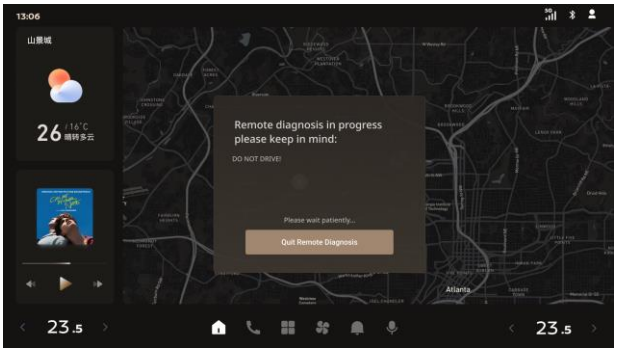

Figure 21. HMI Simulator UI: Active Remote Diagnostic Session

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

VII. CONCLUSION

Similar to our previous work, the communication software to the remote operator, "Public SOVD Server" in SOVD case, must be isolated from the security critical software modules "Authorization Server" and "SOVD manager". This measure will help to mitigate risks caused by implementation of HTTPS function in HPC.

The most important point of security-risk-mitigations for SOVD remote diagnosis is the privilege authorization in the vehicle cabin, because the feasibility of cyber-attack to the remote diagnosis system becomes easy for the attacker if both of user authentication and privilege authorization are possible on the public network. The authorization operation by in-vehicle HMI device can proof that VO (or a local maintenance operator trusted by VO) authorizes the necessary privileges for its requested remote diagnosis.

The second important point is having the proxy server between the vehicle and public network to hide the URL of in-vehicle HTTPS server. It can make difficult the port scanning by attackers and avoid unexpected access to open port 443 for HTTPS communication.

We conclude that these additional security-risk-mitigations can reduce cyber-attack feasibility to the remote diagnosis on SOVD systems.

ACKNOWLEDGMENT

We thank Nissan Technology Development Shanghai to make and test our PoC environment for this research.

REFERENCES

[1] M. Miyashita and H. Takakura "Security-risk-mitigation Measures for Automotive Remote Diagnostic System". In Proceedings of the Eighteenth International Conference on Emerging Security Information, Systems, and Technologies (SECURWARE 2024), Nice, France. 2024.

[2] C. Miller and C. Valasek, " Remote Exploitation of an Unaltered Passenger Vehicle", pp. 84–85, Blackhat Aug. 2015.

[3] H. Wen, Q. A. Chen and Z. Lin, "Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A NewOver-the-Air Attack Surface in Automotive IoT", pp. 960–961, Aug. 2020.

[4] Official Journal of the European Union. (2018). Regulation (EU) 2018/858, OJ L 151, 14.6.2018, p. 1.

[5] S. Robert and J.S. Jayasudha, "Overview of Diagnostic over IP (DOIP), Ethernet Technology and Lightweight TCP/IP for Embedded System", International Journal of Advanced Research in Computer Science, pp. 296–299, 2013.

[6] R. B. Gujanatti, S. A. Urabinahatti and M. R. Hudagi, "Suvey on Security Aspects Related to DoIP", International Research Journal of Engineering and Technology, pp. 2350–2355, 2017.

[7] M. Matsubayashi et al., "Attacks Against UDS on DoIP by Exploiting Diagnostic Communications and Their Countermeasures", 2021 IEEE 93rd Vehicular Technology Conference, pp. 1922–1927, 2021.

[8] ASAM e.V., ASAM SOVD v1.0.0: (Service-Oriented Vehicle Diagnostics). https://www.asam.net/standards/detail/sovd/ (Accessed: July. 22, 2025).

[9] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," in Blackhat USA. Las Vegas, NV, USA: Blackhat Press, pp. 86-88, 2015.

[10] S. Kulandaivel, "Revisiting remote attack kill-chains on modern invehicle networks," PhD thesis, Carnegie Mellon University, pp. 28, 2021.

[11] K. Daimi, "A Security Architecture for Remote Diagnosis of Vehicle Defects", The Thirteenth Advanced International Conference on Telecommunications, pp. 1-7, 2017.

[12] N. Sakimura, J. Bradley, M. Jones, B. De Medeiros, and C. Mortimore, "Openid connect core 1.0," The OpenID Foundation, p. S3, 2014.

[13] D. Hardt, "The OAuth 2.0 Authorization Framework," Internet Requests for Comments, RFC Editor, RFC 6749, October 2012. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6749.txt. (Accessed: July. 22, 2025).

2025, © Copyright by authors, Published under agreement with IARIA - www.iaria.org