2024 IEEE International Symposium on Systems Engineering (ISSE) | 979-8-3503-5372-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISSE63315.2024.10741089

A Systems Approach for Designing Open Vehicle Data Archiving Systems

Chandrima Ghatak
Department of Systems Engineering
Colorado State University
Fort Collins, USA
chandrima.ghatak@colostate.edu

Rik Chatterjee
Department of Systems Engineering
Colorado State University
Fort Collins, USA
rik.chatterjee@colostate.edu

Martin ”Trae” Span
Department of Systems Engineering
Colorado State University
Fort Collins, USA
trae.span@colostate.edu

Jeremy Daily
Department of Systems Engineering
Colorado State University
Fort Collins, USA
jeremy.daily@colostate.edu

***Abstract*—The automotive ecosystem has rapidly transformed through the integration of digital systems, positioning the sector at the forefront of operational technology research and development. As vehicular technologies evolve, the demand for comprehensive and actionable automotive datasets has become critical. These datasets are essential not only for understanding the intricacies of modern automotive technologies but also for developing tools that enhance the safety and functionality of vehicles. However, current data collection tools often fail to capture the multi-layered complexity of automotive data, leading to datasets that are fragmented and limited in scope. This restriction impedes deep analytical insights and subsequent advancements in the field. Moreover, these tools frequently lack robust security measures, compromising the integrity and confidentiality of critical data.**

To address these issues, a systems approach is necessary, We propose a secure data archiving device, designed with a strong emphasis on systems security engineering utilizing a Model-Based Systems Engineering (MBSE) approach. This approach ensures systematic and integrated data capture at every level, from basic vehicle operations to more complex interactions. Our systems security oriented strategy not only preserves the integrity and confidentiality of the data but also provides a holistic view of vehicular networks, which is crucial for advancing research in areas heavily impacted by cybersecurity concerns. Designed from the ground up as an open-source solution, our device challenges the notion that open systems cannot be secure and demonstrates how systematic planning can lead to superior data collection capabilities. By leveraging advanced systems engineering techniques, our device sets the stage for significant improvements in vehicle system research, particularly in domains affected by cybersecurity and intelligent transportation challenges.

***Index Terms*—Security, Data, MBSE, Automotive**

I. Introduction

The electronification of modern vehicles has transformed the automotive ecosystem, ushering in vast improvements in safety, performance, and functionality. This digitization has not only enhanced vehicle systems but has also placed the field of automotive technology at the front line of operational technology research and development. These advancements are inherently data-driven, making automotive datasets a highly valuable commodity for developing tools that further enhance vehicle safety and functionality.

Despite the critical need for comprehensive automotive data, the availability of such datasets is sparse, and those that are available often fail to capture a detailed and holistic picture of modern vehicle operations. This deficiency can primarily be attributed to the limitations of current tools used for capturing automotive data. These tools are generally designed for specialized use cases or are capable of capturing only dispersed data, leaving significant gaps in the data collected. Moreover, they tend to be either prohibitively expensive and tailored for specific functionalities or economically priced but severely limited in their data capture capabilities. This creates a critical gap in the market for tools that are cost-effective yet capable of capturing extensive data from modern vehicles. Moreover, the integrity and confidentiality of data are paramount in automotive systems, where vulnerabilities can lead to severe implications, from privacy breaches to safety risks. Current data collection tools often lack robust security measures, highlighting the need for a more integrated, secure approach to data logging in automotive research.

In response to these challenges, we executed a systems approach, developing a secure, open-source data logging device designed with a strong emphasis on systems security engineering. Designed with a systems approach aided by Model-Based Systems Engineering (MBSE), our device ensures systematic and integrated data capture at every level—from basic vehicle operations to more complex interactions. This approach not only preserves the integrity and confidentiality of the data but also provides a holistic view of vehicular networks, which is crucial for advancing research in areas heavily impacted by cybersecurity concerns. Contrary to common misconceptions, our open-source device demonstrates that open development can lead to more secure and robust systems. The transparency of open-source systems allows for continuous peer review and community-driven improvements, which can quickly identify and rectify potential security flaws.

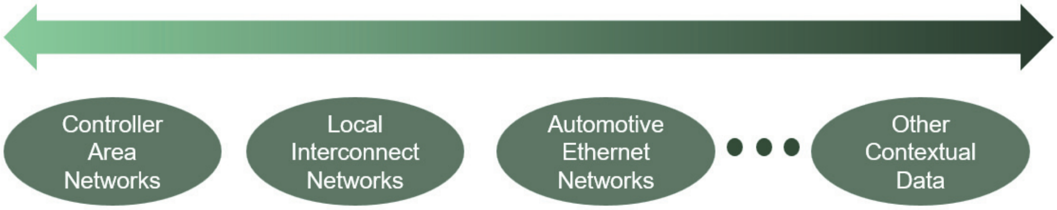


Fig. 1: Spectrum of Automotive Data

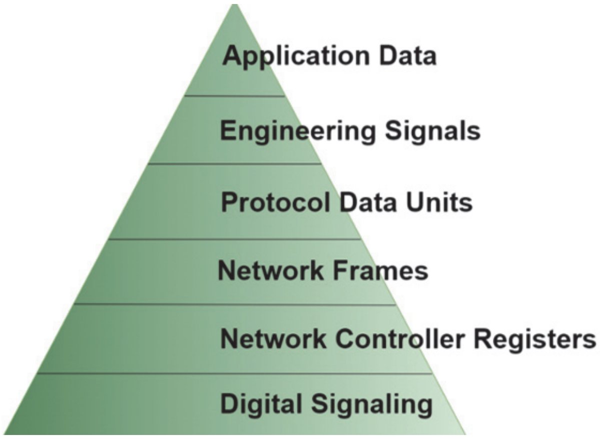


Fig. 2: Layers of Automotive Network Data

The primary objective of this paper is to outline the design and implementation of our device, demonstrating how it addresses the limitations of current tools while enhancing data security and system integration. The paper details the design process, security considerations integrated into each stage of development, and the testing and validation efforts that confirm the effectiveness of the device.

Following this introduction, the paper is organized into sections covering background information on vehicular systems and relevant challenges of data collection in Section II, system context and requirements in Sections III and IV, detailed design and architecture in Section V, implementation in Section VI, and testing and validation in Section VII. Finally, we conclude with a brief discussion of our contributions in Section VIII. Each section builds upon the last, illustrating how a rigorous systems engineering approach can lead to significant advancements in the field of automotive technology research, particularly in areas impacted by cybersecurity and intelligent transportation systems.

II. Background

Modern vehicles are intricate systems, equipped with numerous electronic components and networks. To fully grasp this complexity, we analyze automotive data through two main lenses: depth and breadth. The 'Depth' of automotive data can be visualized as a pyramid, see Figure 2. At its base are digital signals, the fundamental electrical values representing ones and zeros. Moving up, we encounter network controller register data, which controls and monitors the activity of network devices. Next, we find network frames that encapsulate the actual data transmitted across the networks. Above this, Protocol Data Units (PDUs) organize this data into packets for transmission. Nearing the top, engineering signals convert raw data into usable formats for diagnostics and monitoring. At the apex lies application layer data, where data is fully processed and integrated into the vehicle's applications, providing actionable insights for advanced functionalities. In terms of 'Breadth,' automotive data encompasses different types of network architectures—such as Controller Area Networks (CAN) and CAN-FD (Flexible Data Rate) [1], Local Interconnect Networks (LIN) [2], and Automotive Ethernet [3] — which facilitate vehicle communication. It also includes diverse contextual data such as environmental conditions and GPS location, adding layers of complexity and richness to the information gathered from vehicles.

Despite the rich potential data, traditional data capture systems typically focus on specific protocols or data types, often operating in isolation without the capability to integrate data across these multiple sources. This narrow focus results in significant gaps in the collected data, impeding comprehensive analysis. Commercial off-the-shelf (COTS) products, commonly used for data collection, are mostly closed-source and offer limited customization, which restricts their usefulness in academic research [4]–[6]. These tools are often prohibitively expensive for many academic institutions and fail to capture data across all vehicular networks, further narrowing their utility. In the academic realm, numerous efforts have been made to develop tools that can capture data from vehicles [7]–[9]. However, these tools frequently have limited capabilities and struggle to keep pace with modern vehicular architectures. Many are plagued by an inability to capture all necessary data at the high data rates required for meaningful analysis.

Given the rising frequency and sophistication of cyberattacks on embedded systems and vehicles [10]–[13], the security of data capture devices has become increasingly critical with research suggesting different methods to mitigate such risks [14], [15]. Recent studies have highlighted vulnerabilities in Electronic Logging Devices (ELDs), suggesting that without robust security measures, these devices are inherently at risk [16].

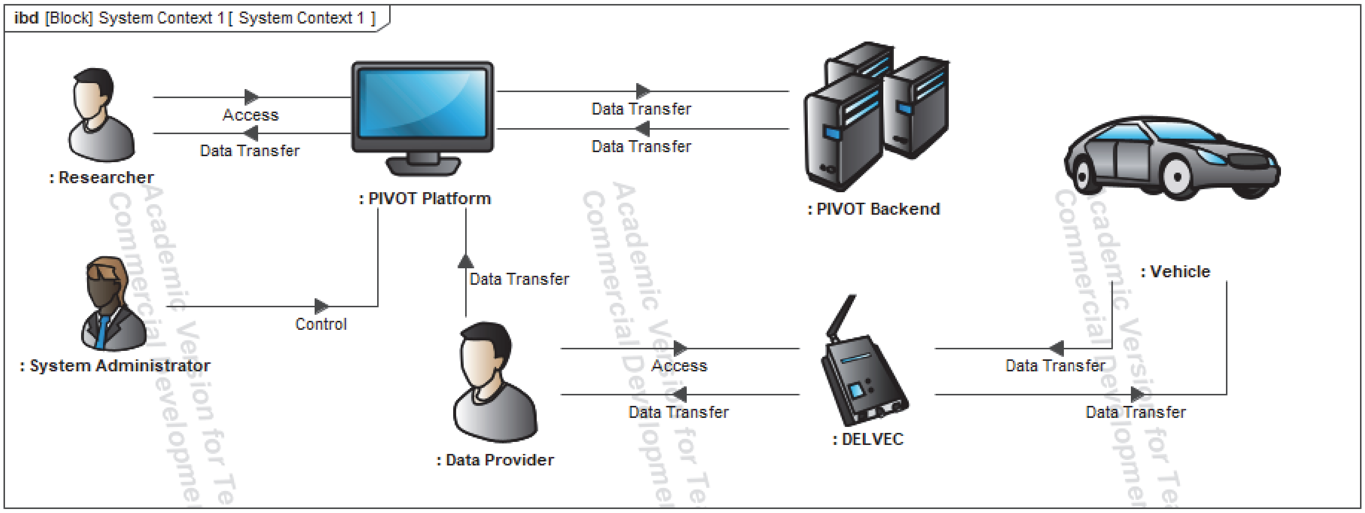


Fig. 3: System Context

III. System Context

This section outlines the context of our data logging device - Device for Layered Vehicular Data Capture (DELVEC), henceforth referred to as DELVEC, within the larger ecosystem of the PIVOT (Platform for Innovative Use of Vehicle Open Telematics) platform [17]. The PIVOT project is designed as a platform for the collection and dissemination of automotive data to enhance vehicular data accessibility and analysis, supporting a wide range of research in areas such as vehicle system cybersecurity, intelligent transportation, and smart connected communities.

The system context diagram (Figure 3) illustrates the central role of DELVEC within the PIVOT ecosystem. As the System of Interest (SOI), DELVEC is engineered to interact directly with vehicles to capture data across various communication protocols and at different layers. It acts as the primary data-acquiring engine for supporting the PIVOT project. The overarching stakeholder needs of the PIVOT project and the role of DELVEC in the overall system are enumerated as follows:

- **Data Collection:** DELVEC collects vehicular data from operational metrics to complex diagnostics, leveraging network protocols such as CAN, CAN-FD, LIN, and Automotive Ethernet.
- **Data Transmission and Security:** After collection, the data is securely transmitted to the PIVOT backend, where it undergoes processing and storage. DELVEC, the Data Provider, and PIVOT Platform ensure that data integrity and security are maintained during transmission, adhering to stringent cybersecurity protocols to safeguard sensitive information.
- **Research Utilization:** Researchers access the data through the PIVOT platform, utilizing DELVEC's data sets for various analytical purposes. This setup facilitates the development of innovative tools and methodologies, pushing the boundaries of current vehicular research.
- **System Administration:** The operation and maintenance of the PIVOT platform are managed by system administrators, who ensure that data flows smoothly from DELVEC to the backend systems and that access is securely controlled.

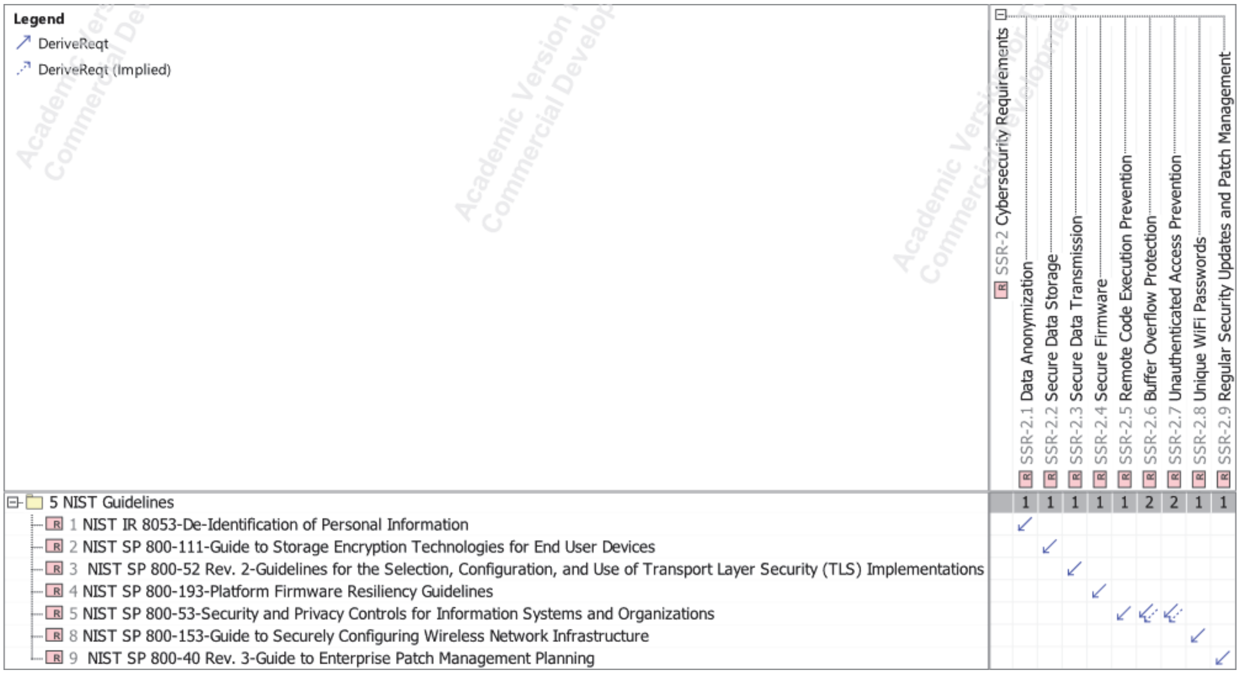
IV. Requirement Analysis

The development of the Device for Layered Vehicular data Capture (DELVEC) necessitates a detailed articulation of system requirements. These requirements form the foundational blueprint guiding all aspects of design, development, and implementation, derived from an in-depth analysis of stakeholder needs, technological capabilities, and industry standards. They ensure that DELVEC meets the current and future demands of vehicular data capture technology.

A. System Requirements

System requirements for DELVEC are designed to ensure compatibility, efficiency, and broad functionality across diverse vehicular environments:

- **SSR-1.1 Compatibility and Interoperability**: The subsystem shall be compatible with a wide range of vehicles, including passenger and commercial medium- and heavy-duty vehicles.
- **SSR-1.2 Data Storage**: The subsystem shall have the capacity to store comparatively large amounts of collected data.
- **SSR-1.3 Data Offloading**: The subsystem shall provide user-friendly means to offload data from the device.
- **SSR-1.4 Types of Data**: The subsystem shall possess the capability to collect data across multiple layers of the vehicle's network protocol.
- **SSR-1.5 Network Protocols**: The subsystem shall support data collection across multiple network protocols such as CAN, CAN-FD, LIN, and Automotive Ethernet.



Legend
↗ DeriveReqt
⇢ DeriveReqt (Implied)

SSR-2 Cybersecurity Requirements	SSR-2.1 Data Anonymization	SSR-2.2 Secure Data Storage	SSR-2.3 Secure Data Transmission	SSR-2.4 Secure Firmware	SSR-2.5 Remote Code Execution Prevention	SSR-2.6 Buffer Overflow Protection	SSR-2.7 Unauthenticated Access Prevention	SSR-2.8 Unique WiFi Passwords	SSR-2.9 Regular Security Updates and Patch Management
5 NIST Guidelines	1	1	1	1	1	2	2	1	1
1 NIST IR 8053-De-Identification of Personal Information	↙								
2 NIST SP 800-111-Guide to Storage Encryption Technologies for End User Devices		↙							
3 NIST SP 800-52 Rev. 2-Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations			↙						
4 NIST SP 800-193-Platform Firmware Resiliency Guidelines				↙					
5 NIST SP 800-53-Security and Privacy Controls for Information Systems and Organizations					↙	⇠	⇠		
8 NIST SP 800-153-Guide to Securely Configuring Wireless Network Infrastructure								↙	
9 NIST SP 800-40 Rev. 3-Guide to Enterprise Patch Management Planning									↙

Fig. 4: Security Requirements mapped to NIST Guidelines

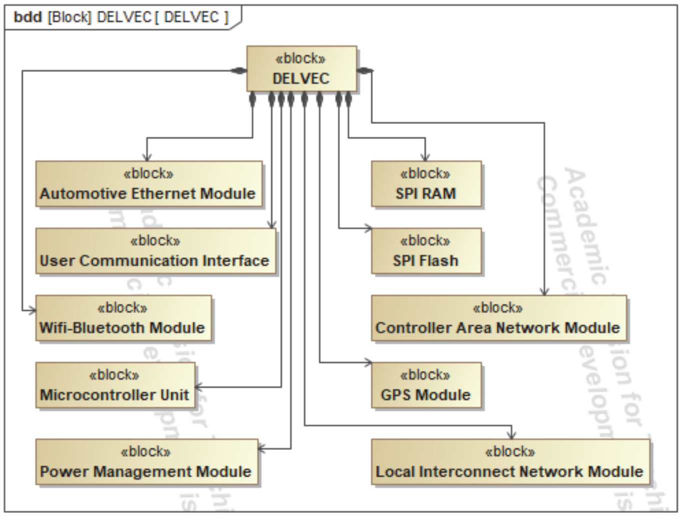
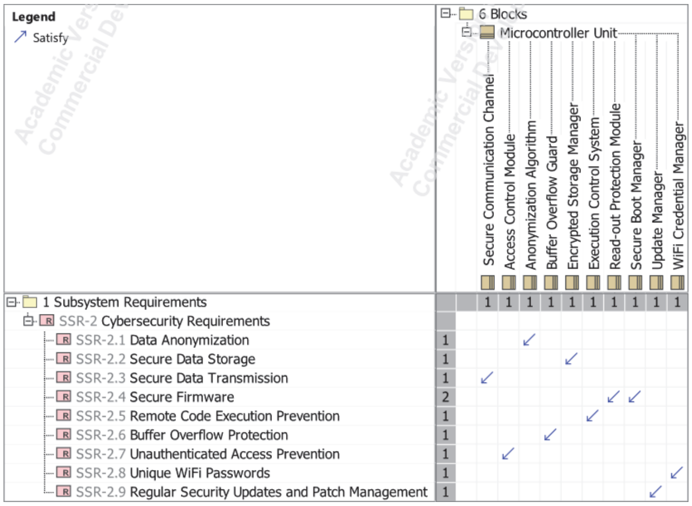

bdd [Block] DELVEC [DELVEC]
«block» DELVEC
«block» Automotive Ethernet Module
«block» User Communication Interface
«block» Wifi-Bluetooth Module
«block» Microcontroller Unit
«block» Power Management Module
«block» SPI RAM
«block» SPI Flash
«block» Controller Area Network Module
«block» GPS Module
«block» Local Interconnect Network Module

Fig. 5: Block Definition Diagram of the DELVEC system



Legend
↗ Satisfy

6 Blocks / Microcontroller Unit	Secure Communication Channel	Access Control Module	Anonymization Algorithm	Buffer Overflow Guard	Encrypted Storage Manager	Execution Control System	Read-out Protection Module	Secure Boot Manager	Update Manager	WiFi Credential Manager
1 Subsystem Requirements	1	1	1	1	1	1	1	1	1	1
SSR-2 Cybersecurity Requirements										
SSR-2.1 Data Anonymization 1			↙							
SSR-2.2 Secure Data Storage 1					↙					
SSR-2.3 Secure Data Transmission 1	↙									
SSR-2.4 Secure Firmware 2							↙	↙		
SSR-2.5 Remote Code Execution Prevention 1						↙				
SSR-2.6 Buffer Overflow Protection 1				↙						
SSR-2.7 Unauthenticated Access Prevention 1		↙								
SSR-2.8 Unique WiFi Passwords 1										↙
SSR-2.9 Regular Security Updates and Patch Management 1									↙	

Fig. 6: Mapping Blocks to Cybersecurity Requirements

- **SSR-1.6 Contextual Data**: The subsystem shall be able to collect other contextual data such as GPS, accurate time-stamps, etc.
- **SSR-1.7 Data Collection Accuracy**: The subsystem shall have the ability to collect data at 100% loads without dropping messages.
- **SSR-1.8 Robustness and Durability**: The subsystem's design and components shall comply with AEC-Q100 Grade 2 automotive reliability standards, ensuring operation within a temperature range of -40°C to +105°C.
- **SSR-1.9 Open Source**: The subsystem's hardware and software designs shall be open-source, supporting customizations and community-driven enhancements.

B. Security Requirements

Ensuring robust security is paramount in the development of DELVEC. Our security requirements and considerations are rigorously derived from the NIST Special Publication 800 series guidelines [18], which set a benchmark for industry compliance in cybersecurity practices.

- **SSR-2.1 Data Anonymization**: The subsystem shall implement data anonymization for Personally Identifiable Information (PII) to de-identify personal information.
- **SSR-2.2 Secure Data Storage**: The subsystem shall employ cryptographic methods to protect stored data.

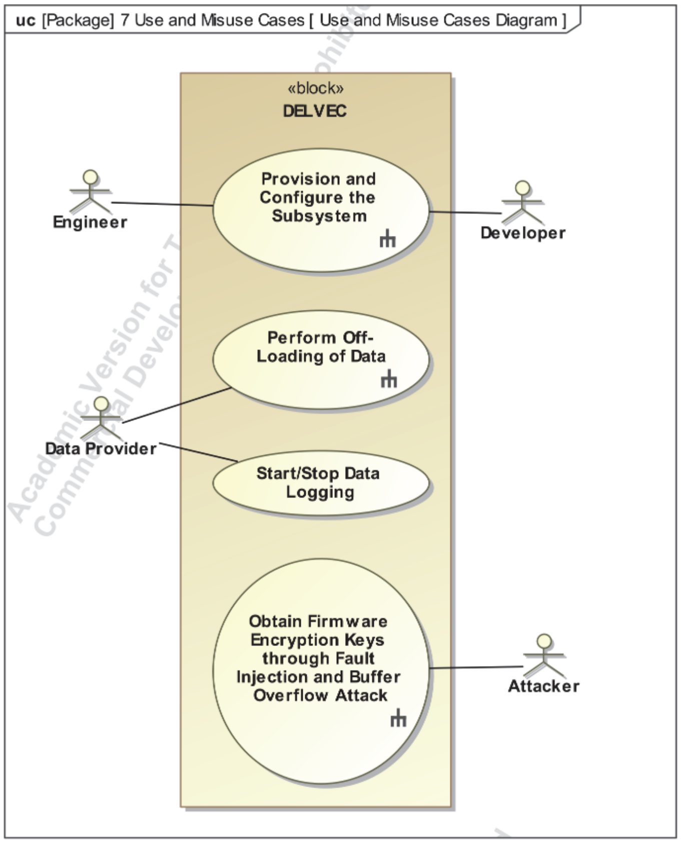

uc [Package] 7 Use and Misuse Cases [Use and Misuse Cases Diagram]
«block»
DELVEC
Provision and Configure the Subsystem
Engineer
Developer
Perform Off-Loading of Data
Data Provider
Start/Stop Data Logging
Obtain Firmware Encryption Keys through Fault Injection and Buffer Overflow Attack
Attacker

Fig. 7: Use and Misuse Case Diagram

- **SSR-2.3 Secure Data Transmission**: The subsystem shall ensure secure data transmission by implementing Transport Layer Security (TLS).
- **SSR-2.4 Secure Firmware**: The subsystem shall have a secure boot process with firmware read-out protection.
- **SSR-2.5 Remote Code Execution Prevention**: The subsystem shall implement runtime protections to prevent remote code execution attacks.
- **SSR-2.6 Buffer Overflow Protection**: The subsystem shall incorporate mechanisms to protect against buffer overflow attacks.
- **SSR-2.7 Unauthenticated Access Prevention**: The subsystem shall require authentication for all access attempts to its interfaces.
- **SSR-2.8 Unique WiFi Passwords**: The subsystem shall enforce unique WiFi passwords for secure network configuration.
- **SSR-2.9 Regular Security Updates and Patch Management**: The subsystem shall maintain a consistent and timely security patch management process.

In Fig. 4, we demonstrate the alignment of our cybersecurity requirements with the NIST guidelines, ensuring that our security practices maintain consistent compliance with established industry standards throughout the development process.

V. Detailed System Modeling

This section explores the architectural design and functional components of the DELVEC, focusing on its modular and secure design tailored for comprehensive vehicular data collection.

A. System Architecture

DELVEC's architecture supports various vehicular data protocols including CAN, CAN-FD, LIN, and Automotive Ethernet, making it highly adaptable to diverse research needs and technological advancements. The core of DELVEC is a multi-protocol data logger capable of high throughput and minimal latency, essential for real-time data capture.

Key Components Include:

- **Microcontroller Unit (MCU)**: Acts as the central processing unit, coordinating the operations of DELVEC.
- **Communication Modules**: Includes WiFi-Bluetooth for wireless communication, GPS for location data, and modules for Automotive Ethernet, CAN and LIN protocols, facilitating extensive data handling capabilities.

The modularity of DELVEC is depicted in the Block Definition Diagram (BDD) shown in Figure 5. Each component's design incorporates specific security measures mapped against our stringent cybersecurity requirements, enhancing DELVEC's overall security posture. This mapping is illustrated in a Satisfy Requirement Matrix in Figure 6.

B. Use and Misuse Case Analysis

While defining use cases for our system, we added the consideration of misuse cases to identify potential threats during the development phase itself. Considering an attacker as a potential actor in our system, we could identify security vulnerabilities early on. The use and misuse case diagram, shown in Figure 7 visualizes normal operations and potential security threats, aiding in a comprehensive understanding of DELVEC's operational and security dynamics.

C. Detailed Activity Diagrams

For an in-depth look at operational processes and security mechanisms, activity diagrams for both a standard use case and a misuse case are provided in Figures 9 and 10 respectively. These diagrams detail the step-by-step activities, highlighting the security checks and data flow controls implemented throughout DELVEC's operations.

D. State Machine Diagram

Finally, the state machine diagram shown in Figure 8 illustrates the various states of DELVEC during operation, showcasing how it handles transitions between different operational and error states to maintain data integrity and security.

VI. Design Implementation

The implementation of DELVEC involves both a hardware and a software element, tailored to enhance the system's robust data capture capabilities. This section details the critical components of the implementation, focusing on both hardware design and software programming.

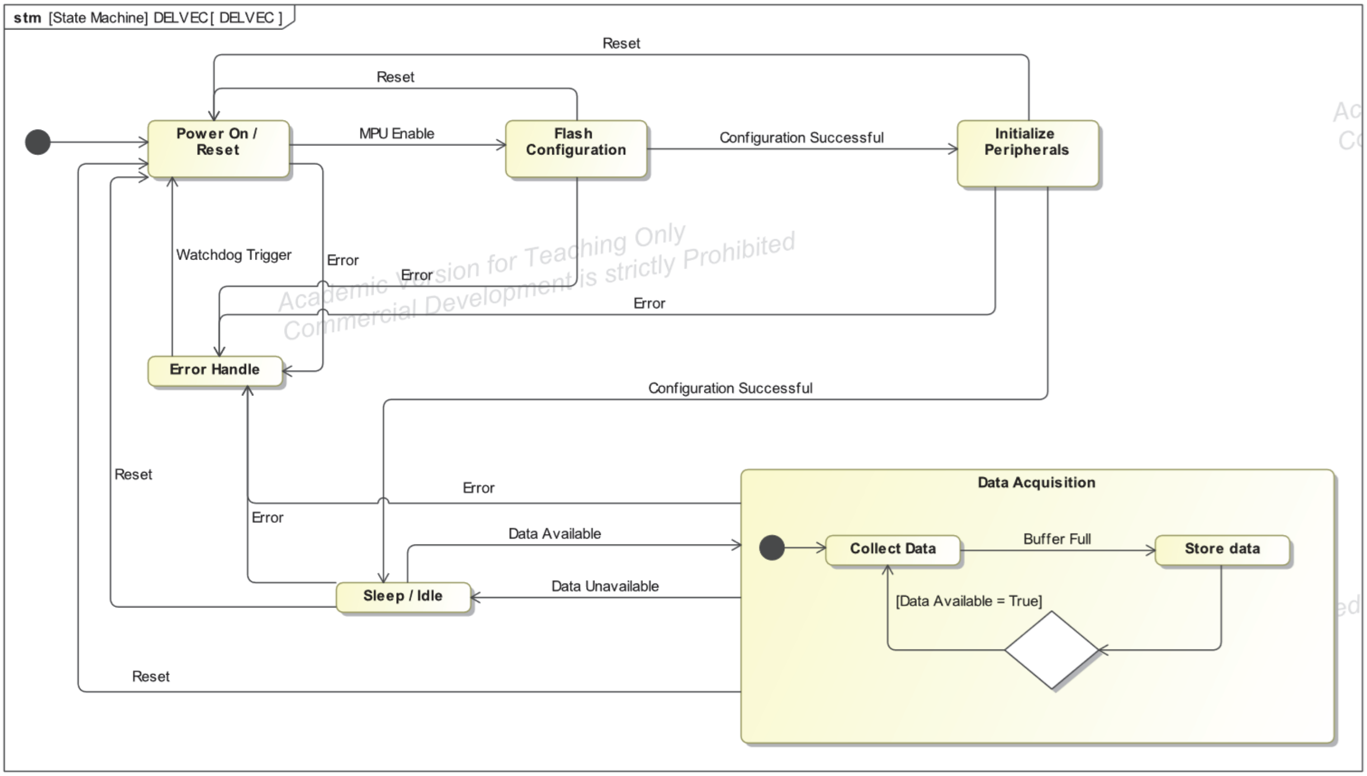


Fig. 8: State Machine Diagram of the DELVEC system

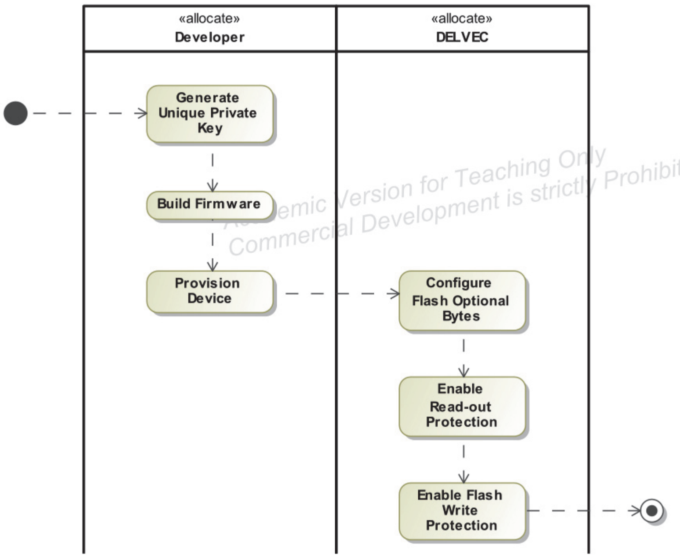

«allocate» Developer
«allocate» DELVEC
Generate Unique Private Key
Build Firmware
Provision Device
Configure Flash Optional Bytes
Enable Read-out Protection
Enable Flash Write Protection

Fig. 9: Activity Diagram for Initial Provisioning

A. Hardware Design

The hardware of DELVEC is engineered to facilitate multi-protocol data logging, which is vital for interfacing with diverse vehicular networks. Key components of this design, as previously discussed in Section V, support a wide array of functionalities. We utilized Altium Designer, a leading PCB

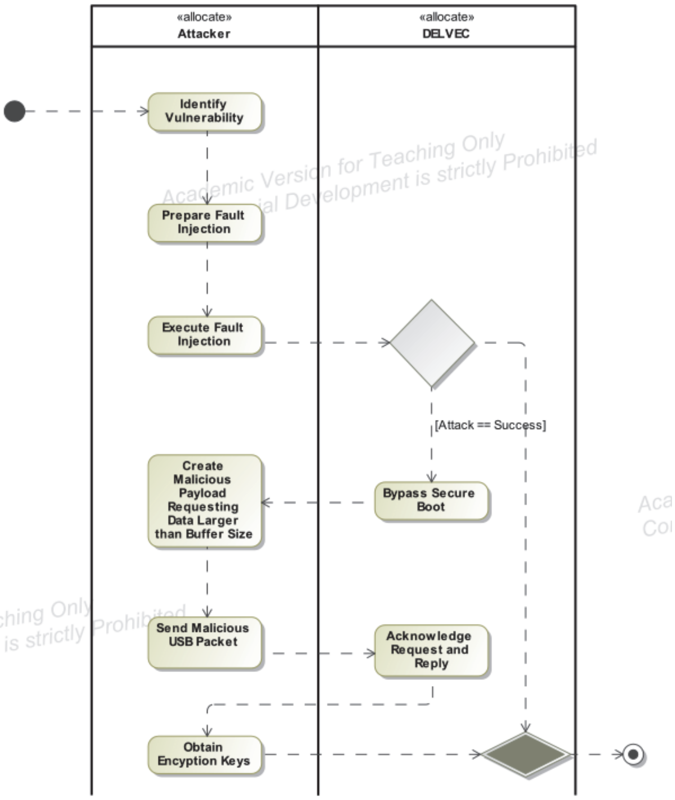


Fig. 10: Activity Diagram Demonstrating Misuse Case

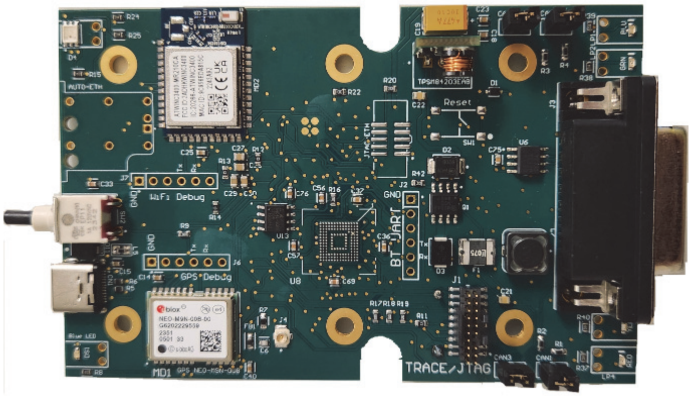
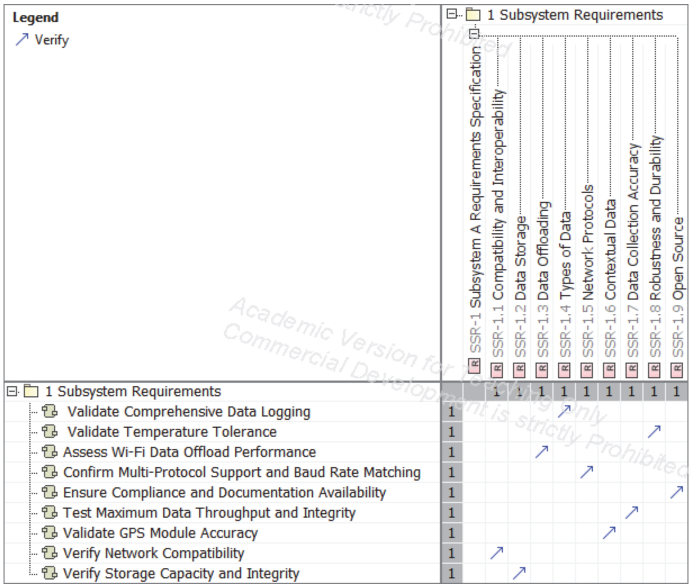


Fig. 11: Prototype of DELVEC



Legend
↗ Verify

		SSR-1 Subsystem A Requirements Specification	SSR-1.1 Compatibility and Interoperability	SSR-1.2 Data Storage	SSR-1.3 Data Offloading	SSR-1.4 Types of Data	SSR-1.5 Network Protocols	SSR-1.6 Contextual Data	SSR-1.7 Data Collection Accuracy	SSR-1.8 Robustness and Durability	SSR-1.9 Open Source
1 Subsystem Requirements			1	1	1	1	1	1	1	1	1
Validate Comprehensive Data Logging	1					↗					
Validate Temperature Tolerance	1									↗	
Assess Wi-Fi Data Offload Performance	1				↗						
Confirm Multi-Protocol Support and Baud Rate Matching	1						↗				
Ensure Compliance and Documentation Availability	1										↗
Test Maximum Data Throughput and Integrity	1								↗		
Validate GPS Module Accuracy	1							↗			
Verify Network Compatibility	1		↗								
Verify Storage Capacity and Integrity	1			↗							

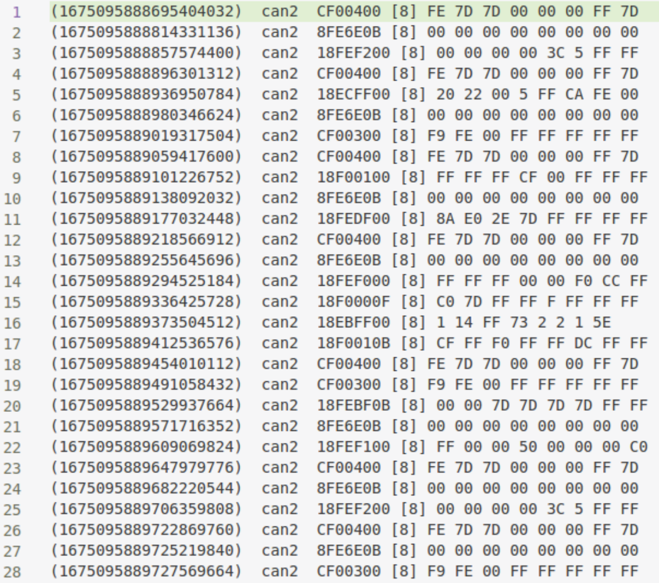
Fig. 12: Mapping of Test Cases to System Requirements

design software, to develop our hardware from the initial concept to the final printed circuit board (PCB) layout. Altium Designer's extensive library of components and their footprints significantly streamlined our design process, enabling efficient integration and layout optimization. The prototype design, showcased in Fig 11, illustrates the sophisticated circuitry that underpins the functionality of DELVEC. Altium's advanced routing capabilities and design automation tools were instrumental in achieving an optimized, reliable hardware design.

The hardware components include a central STM32 Microcontroller Unit with additional components that support different network protocols, data storage capabilities and user interface modules.

B. Software Architecture

The software for DELVEC is prototyped using the STM development tools. Key components of the software include:



```
(1675095888695404032)  can2  CF00400 [8] FE 7D 7D 00 00 00 FF 7D
(1675095888814331136)  can2  8FE6E0B [8] 00 00 00 00 00 00 00 00
(1675095888857574400)  can2  18FEF200 [8] 00 00 00 00 3C 5 FF FF
(1675095888896301312)  can2  CF00400 [8] FE 7D 7D 00 00 00 FF 7D
(1675095888936950784)  can2  18ECFF00 [8] 20 22 00 5 FF CA FE 00
(1675095888980346624)  can2  8FE6E0B [8] 00 00 00 00 00 00 00 00
(1675095889019317504)  can2  CF00300 [8] F9 FE 00 FF FF FF FF FF
(1675095889059417600)  can2  CF00400 [8] FE 7D 7D 00 00 00 FF 7D
(1675095889101226752)  can2  18F00100 [8] FF FF FF CF 00 FF FF FF
(1675095889138092032)  can2  8FE6E0B [8] 00 00 00 00 00 00 00 00
(1675095889177032448)  can2  18FEDF00 [8] 8A E0 2E 7D FF FF FF FF
(1675095889218566912)  can2  CF00400 [8] FE 7D 7D 00 00 00 FF 7D
(1675095889255645696)  can2  8FE6E0B [8] 00 00 00 00 00 00 00 00
(1675095889294525184)  can2  18FEF000 [8] FF FF FF 00 00 F0 CC FF
(1675095889336425728)  can2  18F0000F [8] C0 7D FF FF F FF FF FF
(1675095889373504512)  can2  18EBFF00 [8] 1 14 FF 73 2 2 1 5E
(1675095889412536576)  can2  18F0010B [8] CF FF F0 FF FF DC FF FF
(1675095889454010112)  can2  CF00400 [8] FE 7D 7D 00 00 00 FF 7D
(1675095889491058432)  can2  CF00300 [8] F9 FE 00 FF FF FF FF FF
(1675095889529937664)  can2  18FEBF0B [8] 00 00 7D 7D 7D 7D FF FF
(1675095889571716352)  can2  8FE6E0B [8] 00 00 00 00 00 00 00 00
(1675095889609069824)  can2  18FEF100 [8] FF 00 00 50 00 00 00 C0
(1675095889647979776)  can2  CF00400 [8] FE 7D 7D 00 00 00 FF 7D
(1675095889682220544)  can2  8FE6E0B [8] 00 00 00 00 00 00 00 00
(1675095889706359808)  can2  18FEF200 [8] 00 00 00 00 3C 5 FF FF
(1675095889722869760)  can2  CF00400 [8] FE 7D 7D 00 00 00 FF 7D
(1675095889725219840)  can2  8FE6E0B [8] 00 00 00 00 00 00 00 00
(1675095889727569664)  can2  CF00300 [8] F9 FE 00 FF FF FF FF FF
```

Fig. 13: Snapshot of Data Logged

- **Real-time Data Processing**: Software algorithms are optimized for real-time data processing and logging.
- **Data Encryption and Security Protocols**: To ensure the security of data transmission, all data is encrypted using advanced encryption standards. Additional security measures are implemented to prevent unauthorized data access.
- **User Interface**: A user interface is provided for system configuration and monitoring.

The successful implementation of DELVEC, shown in Fig 13 shows a capture of CAN data using a development board and demonstrates its capability to serve as a comprehensive tool for vehicular data capture, providing valuable insights into vehicle operations and performance.

VII. Testing and Validation

The testing and validation phase of the DELVEC system was meticulously planned and executed to ensure that the device meets all predefined requirements. This phase was critical in demonstrating the functionality, performance, and security of the system. Testing was structured around three main objectives:

1) **Functionality Testing**: To ensure that DELVEC operates as intended, supporting various vehicular data protocols and accurately capturing comprehensive vehicular data.
2) **Performance Testing**: To assess the system's data throughput, latency, and reliability under typical operating conditions.
3) **Security Testing**: To validate the effectiveness of security measures implemented within DELVEC, ensuring the protection of data against unauthorized access and cyber threats.

Each test case was designed to verify specific system requirements and mapped against these objectives to demonstrate compliance with the project's standards and stakeholder expectations. The verification of each requirement was supported by detailed test cases. These cases were methodically linked to specific requirements through a verification matrix, as shown in Figure 12, to ensure comprehensive coverage and traceability.

A key aspect of the validation process involved testing the device's ability to capture CAN data from a vehicle in real-time. The test successfully demonstrated DELVEC's capacity to interface with vehicle networks and accurately log data, confirming the functionality and performance of the prototype. This successful test is a significant milestone, verifying that DELVEC meets the critical operational requirement of real-time data logging from complex vehicular networks.

VIII. Conclusion

This research effort has demonstrated the utility of a systems approach with a focus on systems security engineering for designing secure vehicular data capture systems. Through this structured systems approach, we meticulously defined and aligned system requirements with detailed Model-Based Systems Engineering (MBSE) artifacts to construct a robust architecture. This process ensured that security was not an afterthought but a fundamental aspect of design, effectively integrating cybersecurity measures with system functionality. The resulting DELVEC system not only meets diverse vehicular data capture needs but also adheres to high standards of data integrity and security, showcasing the critical role of systems security engineering in enhancing traditional engineering practices within complex system domains.

Acknowledgment

This material is based upon work supported by the National Science Foundation (NSF) under Contract No. 2213735. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

References

[1] Robert Bosch GmbH, "CAN Specification," Robert Bosch GmbH, Standard 2.0, 1991.

[2] International Organization for Standardization, "ISO 17987-7:2016, Road vehicles - Local Interconnect Network (LIN) - Part 7: Electrical Physical Layer (EPL) conformance test specification," https://www.iso.org/standard/61189.html, 2016, accessed: 2023-12-27.

[3] "IEEE Standard for Ethernet Amendment 4: physical layer specifications and management parameters for 1 Gb/s operation over a single twisted-pair copper cable," *IEEE Std 802.3bp-2016 (Amendment to IEEE Std 802.3-2015 as amended by IEEE Std 802.3bw-2015, IEEE Std 802.3by-2016, and IEEE Std 802.3bq-2016)*, pp. 1–211, 2016.

[4] Vector Informatik, "Smart Logger - Data Logger for CAN, CAN FD, LIN, FlexRay, and Automotive Ethernet," https://www.vector.com/us/en/products/products-a-z/hardware/data-logger/smart-logger/c322811, 2023, accessed: 2023-12-28.

[5] CSS Electronics, "CANedge2: CAN Bus Data Logger with WiFi," https://www.csselectronics.com/products/can-bus-data-logger-wifi-canedge2, 2023, accessed: 2023-12-28.

[6] Intrepid Control Systems, "ValueCAN 4: Vehicle Network Adapter," https://intrepidcs.com/products/vehicle-network-adapters/valuecan-4/, 2023, accessed: 2023-12-28.

[7] D. Van, "Secure can logging and data analysis," Master's Thesis, Colorado State University, Fort Collins, Colorado, 2020, accessed on 2023-12-28. [Online]. Available: https://mountainscholar.org/items/e7247782-3389-4505-8433-85eb34950aa0

[8] M. Johanson and L. Karlsson, "Improving vehicle diagnostics through wireless data collection and statistical analysis," in *2007 IEEE 66th Vehicular Technology Conference*, Baltimore, MD, 2007.

[9] A. Shaout, D. Mysuru, and K. Raghupathy, "Can sniffing for vehicle condition, driver behavior analysis and data logging," in *2018 International Arab Conference on Information Technology (ACIT)*, Werdanye, Lebanon, 2018.

[10] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," in *USENIX Security Symposium*, vol. 4. San Francisco, CA, USA: USENIX Association, 2011, pp. 447–462.

[11] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," in *Blackhat USA*. Las Vegas, NV, USA: Blackhat Press, 2015.

[12] R. Chatterjee, S. Mukherjee, and J. Daily, "Exploiting transport protocol vulnerabilities in SAE J1939 networks," in *Proceedings of the Inaugural International Symposium on Vehicle Security & Privacy*. San Diego, CA, USA: Internet Society, 2023. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2023/02/vehiclesec2023-23053-paper.pdf

[13] R. Chatterjee, C. Green, and J. Daily, "Exploiting diagnostic protocol vulnerabilities on embedded networks in commercial vehicles," in *Symposium on Vehicles Security and Privacy (VehicleSec)*. San Diego, CA, USA: NDSS Symposium, 2024. [Online]. Available: https://dx.doi.org/10.14722/vehiclesec.2024.23046

[14] C. Ghatak, S. Jabeen, H. Shirazi, and I. Ray, "Improving the resiliency of embedded networks in heavy vehicles - towards fault tolerance," in *Proceedings of Ninth Annual Industrial Control System Security (ICSS) Workshop*. Annual Computer Security Applications Conference (ACSAC), 2023. [Online]. Available: https://www.acsac.org/2023/workshops/icss/chandrima-ghatak-paper.pdf

[15] R. Chatterjee, B. Karel, R. Baratto, M. Gordon, and J. Daily, "Assured micropatching of race conditions in legacy real-time embedded systems," in *Proceedings of the Conference on Real-Time Embedded Systems*, 2024.

[16] J. Jepson, R. Chatterjee, and J. Daily, "Commercial vehicle electronic logging device security: Unmasking the risk of truck-to-truck cyber worms," in *Symposium on Vehicles Security and Privacy (VehicleSec)*. San Diego, CA, USA: NDSS Symposium, 2024. [Online]. Available: https://dx.doi.org/10.14722/vehiclesec.2024.23047

[17] "Collaborative research: Ccri: New: Open community platform for sharing vehicle telematics data for research and innovation," National Science Foundation.

[18] National Institute of Standards and Technology, "NIST Special Publication 800 Series," [Online; accessed 10-May-2024]. [Online]. Available: https://csrc.nist.gov/publications/sp800