Addressing automotive cybersecurity risks with an ARM Morello capability-enhanced prototype

Hoang Nga Nguyen[1], Siraj Ahmed Shaikh[1], Emirhan Kutsal[1], Anna Stylianou[2], Rob Potter[3], and Thomas Sors[3]

[1]Swansea University
[2]Applus+ IDIADA
[3]Beam Connectivity Limited

April 25, 2024

Abstract

Modern vehicles are equipped with hundreds of ECUs deployed across vehicle networks. Each ECU runs a variety of safety and cyber-critical workloads which is facing an increasingly challenging cybersecurity climate, which is being driven by various factors such as vehicle system complexity, software complexity, supply chain complexity and an increase in wireless interfaces. With these challenges, the automotive sector is a perfect domain for the use of Security-by-Design and secure hardware technologies. In this paper, we demonstrate the application of Security-by-Design and the secure hardware Morello, one of the CHERI implementations, in the design and implementation of Telematic Control Units (TCU), a crucial component of modern vehicles. This results in a generic secure TCU design with well-justified security requirements and an understanding of the risks associated with it. As such, this work paves the way for the systematic integration of secure hardware for cyber-physical systems, of which automotive is just one application.

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

STANDARDS

Addressing automotive cybersecurity risks with an ARM Morello capability-enhanced prototype

Hoang Nga Nguyen[1] | **Siraj Ahmed Shaikh**[1] | **Emirhan Kutsal**[1] | **Anna Stylianou**[2] | **Rob Potter**[3] | **Thomas Sors**[3]

[1]Systems Security Group, Swansea University, Swansea, UK

[2]Applus+ IDIADA, Cambridge, UK

[3]Beam Connectivity Limited, Cirencester, UK

Correspondence
Corresponding author Hoang Nga Nguyen.
Email: h.n.nguyen@swansea.ac.uk

Abstract

Modern vehicles are equipped with hundreds of ECUs deployed across vehicle networks. Each ECU runs a variety of safety and cyber-critical workloads which is facing an increasingly challenging cybersecurity climate, which is being driven by various factors such as vehicle system complexity, software complexity, supply chain complexity and an increase in wireless interfaces. With these challenges, the automotive sector is a perfect domain for the use of Security-by-Design and secure hardware technologies. In this paper, we demonstrate the application of Security-by-Design and the secure hardware Morello, one of the CHERI implementations, in the design and implementation of Telematic Control Units (TCU), a crucial component of modern vehicles. This results in a generic secure TCU design with well-justified security requirements and an understanding of the risks associated with it. As such, this work paves the way for the systematic integration of secure hardware for cyber-physical systems, of which automotive is just one application.

KEYWORDS
Automotive Cyber-security, Digital Security by Design, Threat Analysis and Risk Assessment, CHERI, ARM Morello

1 | INTRODUCTION

Traditional vehicle electronic architectures tend towards the use of one Electronic Control Unit (ECU) per vehicle function, which has resulted in a proliferation of ECUs being deployed across the vehicle network. This means that modern luxury vehicles can have as many as 150 separate ECUs, each running a variety of safety and cyber-critical workloads [6]. The automotive industry is facing an increasingly challenging cybersecurity climate, which is being driven by a number of factors:

Vehicle system complexity: The shift to Electric Vehicles (EVs) has led to a ground-up rethink of vehicle electronic architectures and an opportunity to leverage more powerful system-on-chip (SoC) modules and hypervisor technology. This has resulted in more centralised, shared compute resources and the co-location of different software functions of varying criticality onto the same SoC. This increase in the depth of the software stack, including a host OS and hypervisor makes the overall system more complex.

Software complexity: The introduction of cellular connectivity facilitates vehicle OEMs in delivering software updates Over-The-Air (OTA) and allowing manufacturers to roll out new software features in the field. This approach is commonplace in consumer electronics devices, but new to the automotive industry. The number of features in the typical vehicle has been steadily increasing as has the share of those implemented just in software, so much so that the term *Software Defined Vehicle* (SDF) has been coined to illustrate this shift in the importance of software in the modern vehicle.

Supply chain complexity: The automotive industry has established a complex supply chain for electronics and software, which has developed due to the specialisation required to build and certify automotive-grade components. This specialisation

means that most vehicle manufacturers are more commonly integrating third-party components rather than developing systems in-house.

Increase in wireless interfaces: All interfaces into a system can become a point of ingress for a cyber attack, but wireless interfaces represent an easier target because they don't require physical access or even proximity to the vehicle. Since 2018, cellular modems have been a regulatory requirement for new vehicles in EU to deliver emergency call capability (eCall) in the event of an accident. Many other wireless protocols are included in the modern vehicle as part of the security, safety and infotainment capabilities. These wireless interfaces include Bluetooth/BLE, Wi-Fi (client, Access Point and Wi-Fi Direct), GNSS (GPS, Galileo, GLONASS, et al), V2X via DSRC/5G, and Ultra-Wide Band (UWB).

Ultimately, this means vehicles have a larger, more complex software stack than ever before, integrating software contributions from dozens of vendors. In addition, they have more points of ingress for cyber attacks and need to maintain secure and resilient operations as any compromises can have serious safety and financial repercussions. Furthermore, the increasing connectivity and autonomy of modern vehicles have led to a growing concern about cybersecurity risks in the automotive industry. The potential impact of cyber attacks on connected and autonomous vehicles could be devastating, ranging from theft and data breaches to physical harm or loss of life. With these challenges, the automotive sector is a perfect domain for the use of Security-by-Design and secure hardware technologies.

On one hand, Security-by-Design can be realised by effective cybersecurity standards and risk assessment methodologies in the automotive industry. Different from the traditional Information Technology systems, it is necessary to consider a suitable approach to planning, implementation and monitoring cyber security due to the high complexity level in the design of connected vehicles [20]. ISO/SAE 21434 [10] is a standard that provides a framework for the development and implementation of cybersecurity measures in road vehicles, while Threat and Risk Assessment (TARA) is a methodology for identifying and assessing potential cybersecurity risks in connected and autonomous vehicles. Together, ISO/SAE 21434 and TARA provide a comprehensive approach to addressing cybersecurity risks in the automotive industry, helping to ensure the safety and security of vehicles and their passengers and the entire vehicle ecosystem. On the other hand, secure hardware provides trusted elements from the design phase where security features are blended into conventional hardware architectures. Leveraging a more secure hardware foundation is a tactic identified by the US Cybersecurity & Infrastructure Agency in their April 2023 whitepaper on Security-by-Design [3], where they recommend CHERI:

> *Incorporate architectural features that enable fine-grained memory protection, such as those described by Capability Hardware Enhanced RISC Instructions (CHERI) that can extend conventional hardware Instruction-Set Architectures (ISAs).*

In this paper, we discuss the application of Security-by-Design and the secure hardware Morello Board, one of the CHERI implementations, in the design and implementation of Telematic Control Units (TCU). In the automotive industry, the TCU is a crucial component. It facilitates wireless communication between vehicles and external systems to provide various services with different latency requirements. These requirements range between hours-minutes-seconds-milliseconds, as listed in the below use cases, respectively:

Vehicle diagnostics data - Processing data from Controller Area Network (CAN), through the TCU and up to the cloud.
OTA software update - Pulling software packages from the cloud, cryptographically verifying them, and passing on other vehicle ECUs.
V2I traffic advisory - Communicating with roadside infrastructure via V2X protocols.
Teleoperation - Operating vehicles at a distance.

As TCU hosts many wireless interfaces including Bluetooth, Wi-Fi, GNSS, V2X via DSRC/5G, and UWB, it provides a wide range of attack surfaces and becomes one of the first targets for compromising and hacking vehicles. Therefore, securing TCU is crucial to protect the safety of vehicles and their users against cyber attacks. In order to solve this problem systematically, it is important to consider security throughout the development cycle of the TCU, starting from the design phase. To this end, we apply Security-by-Design when designing the TCU by performing a rigorous process for Threat Analysis and Risk Assessment (TARA). It enables us to capture possible threats to the TCU as well as to suggest a comprehensive list of security requirements. Among these threats, we show that many are addressed by the use of Morello Boards. Overall, the paper offers the following values to practitioners and researchers in applying TARA for a TCU design using CHERI/Morello:

- A tailored ISO/SAE 21434-compatible approach for Threat Analysis and Risk Assessment (TARA) for designing TCUs. For demonstration purposes, we show the application of this approach to the OTA software update use case. However, it can be employed for other use cases by repeating the same process.
- A comprehensive TARA result for designing TCUs. It includes a list of potential threats towards TCUs and their associated risks. The results highlight that the application of CHERI/Morello Board reduces the risk of memory attacks;
- A comprehensive list of security requirements for TCUs and the associated residual risks.

The paper is organised as follows. In Section 2, we briefly recall background information about CHERI/Morello, ISO21434 and its TARA recommendation, TCUs and OTA software update, and finally how CHERI/Morello can help secure Automotive OTA software update. Then, in Section 3, we discuss our tailored ISO/SAE 21434-compatible approach for Threat Analysis and Risk Assessment (TARA). The result of applying this approach to the OTA software update of TCUs is presented in Section 4. It will also suggest a list of security requirements for this use case. Finally, we discuss the result and conclude the paper in Section 5.

2 | BACKGROUND

In this section, we first begin with some background on TCU and OTA software updates. Then, the secure hardware architecture CHERI and its implementation Morello Board are briefly recalled and their benefits to TCU and OTA software update are discussed. Finally, we focus on the automotive cybersecurity standard ISO/SAE 21434 and its key components in the Threat Analysis and Risk Assessment process.

TCU and Automotive OTA

A Telemantic Unit (TCU) is an onboard computing device. Its main responsibility is to provide a communication service between ECUs within a vehicle with outside services. In other words, it bridges data communication between in-vehicle networks such as CAN and Automotive Ethernet with external ones such as GPS, Wifi, Cellular, etc. For example, engine-related data (engine speed, wheel velocity, oil lever, engine temperatures, . . .) from various ECUs within a vehicle are sent to TCU to forward to a data cloud service via a cellular interface for several purposes such as online diagnostic and insurance. Automotive OTA is another function that is provided by a TCU. ECUs within the vehicle can periodically request new updates to the manufacturer's OTA servers via the TCU. The request is sent to the TCU by an in-vehicle network and forwarded to the OTA servers via a wifi or a cellular interface. In the opposite direction, if there is a new update, firmware can be downloaded to the TCU which will deliver to the relevant ECU. Securing the OTA process is essential to ensure that cyber attackers cannot interfere and compromise ECUs with malicious firmware. Such a malicious firmware could cause catastrophic incidents such as disabling the functionality of the brake ECU while travelling at high speed. To this end, several automotive OTA systems and designs [13, 16, 17, 19] have been proposed. Figure 1 illustrates a canonical design proposed by the Uptane standard [19]. Its operation cycle consists of the following 7 steps:

- Load the current metadata from the storage;
- Send a request to the server (director or image repository) for role metadata;
- Decrypt the signature in the received metadata using the private key;
- Verify the payloads legitimacy from the decrypted signature;
- Ensure that the new metadata version is higher than the old metadata version;
- Check that the current time is lower than the expiration time of the new metadata; and
- Install the new firmware once the verification of the metadata has been confirmed.

The overall concern from these approaches focuses on authentication and encryption aspects to provide security.

CHERI/Morello and its benefits for Automotive OTA

The Morello Board [1] is a prototype development platform that features the CHERI architecture [2, 14]. CHERI's main objective is to develop and deploy capability features in conventional processors. One of them is enhanced memory protection via the

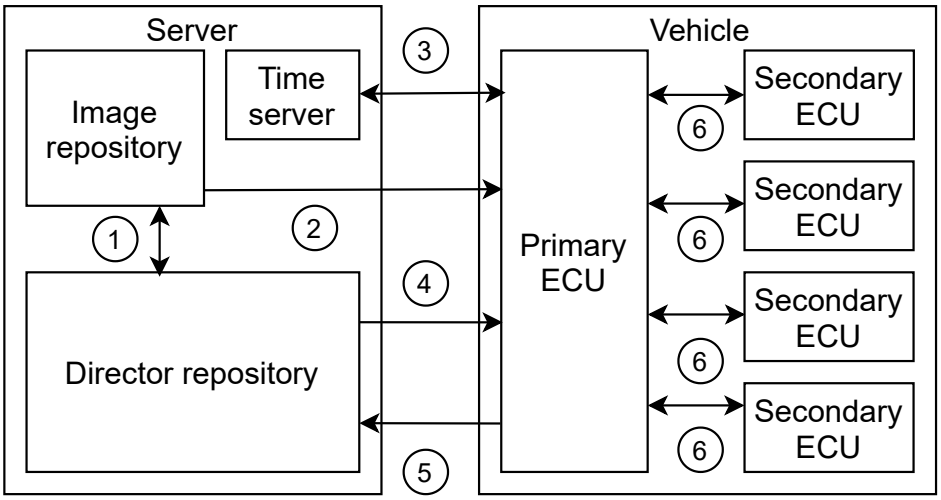

FIGURE 1 A canonical Automotive OTA design proposed by the Uptane standard.

implementation of fine-grained memory management at the hardware level which is adaptable to potentially unsafe memory programming languages such as C/C++. According to a recent Microsoft study [11], CHERI enables us to mitigate at least two-thirds of all memory safety vulnerabilities. The Morello Board also comes with a number of software stacks flavours including bare-metal, Android, Busybox and most recently Debian.

Securing automotive updates is a huge concern because a malicious or corrupt update can lead to severe impacts on connected cars, which could potentially lead to loss of human life [12, 5]. While protection of the OTA updates is vital for any IoT device, update security in connected cars should be an even more significant concern [4]. OEMs must adopt effective security measures for the detection and mitigation/prevention of any potential security breaches in OTA updates. A typical example of such a measure is the hardening of the Telematics Units (TCU) where the OTA functionality is deployed. According to [19], OTA suffers from a number of cyber-attacks including (but not limited to) intercepting network traffic between OTA servers and clients and compromising OTA servers and TCUs running OTA clients. While various software solutions exist for dealing with network interception attacks by increasing secure authentication and encryption, vulnerabilities in TCUs such as buffer overflow require different countermeasures. Otherwise, they are the potential open backdoors to cyberattacks where the OTA process is compromised and malicious software is installed into various embedded components within vehicles. To this end, hardware-based solutions such as the CHERI architecture for TCUs is promising to provide a holistic solution.

ISO/SAE 21434 and the recommended TARA processes

ISO/SAE 21434 [10] is a cybersecurity standard for road vehicles that provides guidelines for the development and implementation of cybersecurity measures throughout a vehicle's life cycle. The standard was published in September 2021 and it aims to help organisations identify and mitigate potential cybersecurity risks associated with connected vehicles, autonomous vehicles, and other advanced vehicle technologies. For example, the German Automotive Association requires that all development products, from 2022, must be subject to a security assessment [15] if they involve external interfaces that influence automotive functional behaviours (on-board communication, diagnostic, etc.). This is to adhere to the requirements from ISO/SAE 2143O and subsequently the UNECE Regulation R155.

ISO/SAE 21434 is a comprehensive standard that covers the entire product life cycle, including the design, development, production, operation, maintenance, and decommissioning phases. It is designed to be scalable and flexible, allowing organisations of all sizes and types to implement cybersecurity measures that are appropriate for their specific needs. It includes a range of cybersecurity-related requirements, including the need for organisations to conduct threat and risk assessments, implement cybersecurity measures based on those assessments, and establish processes for continuous monitoring and improvement.

One of the key components of ISO/SAE 21434 is the Threat and Risk Assessment (TARA) process, which involves identifying and analysing potential cybersecurity threats and vulnerabilities associated with a vehicle's hardware, software, and network

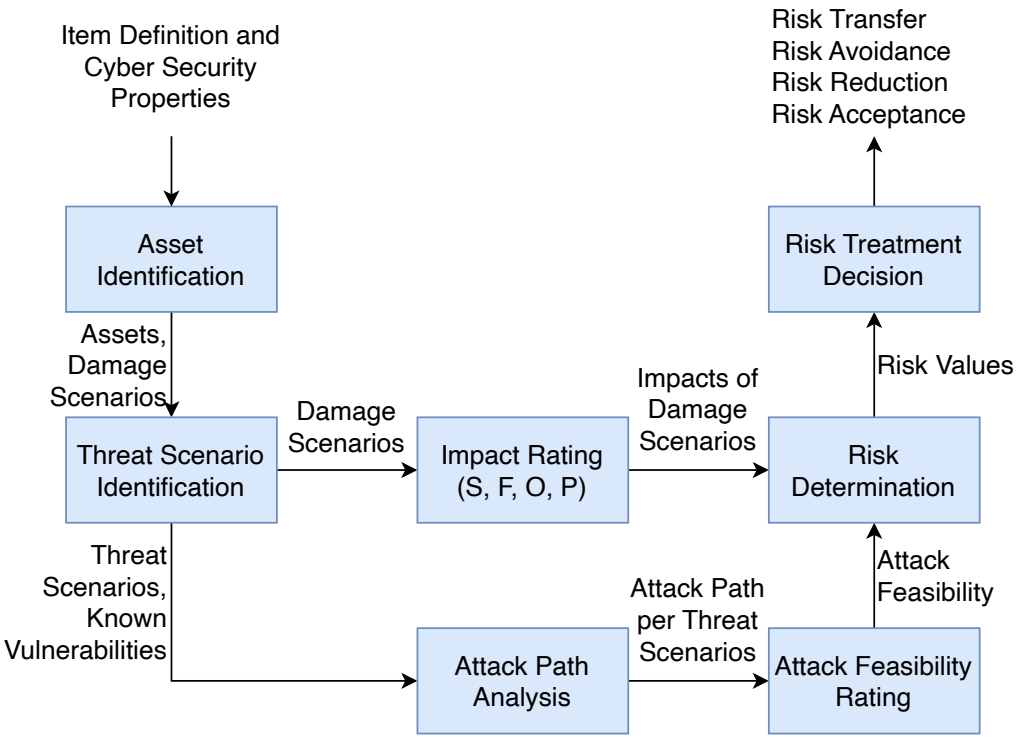

FIGURE 2 Overview of the TARA Methodology.

systems. The TARA process also includes assessing the associated risks associated, deciding a risk treatment and identifying appropriate countermeasures to mitigate those risks.

The TARA methodology presented in this paper is a threat-based methodology used to identify and classify potential threats of the system and ultimately formulate security requirements. This method is fully compliant with ISO/SAE 21434:2021 risk assessment requirements. Repeating the risk assessment should produce consistent results.

Figure 2 illustrates an overview of all the steps taking place in the TARA process. It consists of the following 7 steps:

- Asset Identification. This is the first step of TARA, where assets are enumerated, along with their security properties. This will lead to the damage scenarios.
- Threat Scenario Identification. The threat scenarios for each damage scenario should be identified. A damage scenario shall be associated with at least one threat scenario.
- Impact Rating. The damage scenarios should be assessed against potential adverse consequences for road users in the impact categories of safety, financial, operational, and privacy (SFOP) respectively. The impact rating of a damage scenario should be determined for each impact category to be one of the following: severe, major, moderate or negligible.
- Attack Path Analysis. For each threat scenario, the attack path should be analysed. The attack path analysis can either be based on: 1. top-down approaches that deduce attack paths by analysing the different ways in which a threat scenario could be realised, e.g. attack trees OR 2. bottom-up approaches that build attack paths from the vulnerabilities identified.
- Attack Feasibility Rating. For each attack path, the attack feasibility should be determined. The attack feasibility rating method should be defined based on attack potential-based approach. Attack potential is defined in ISO/IEC 18045 as a measure of the effort to be expended in attacking an item or component, expressed in terms of an attacker's expertise and resources. For the attack potential, the HEAVENS parameters are followed.
- Risk Value Determination. The risk of threat scenarios is determined from the impact associated with its corresponding damage scenario and the likelihood of the associated attack paths.
- Risk Treatment Decision. For each threat scenario, considering the risk values, one or more of the following risk treatment option(s) should be determined:

  · Risk Transfer/Sharing: via suppliers or through insurance.

- Risk Avoidance: by removing risk sources.
- Risk Reduction: by introducing countermeasures.
- Risk Retention/Acceptance: by accepting the risk.

3 | METHODOLOGY

We propose a threat analysis and risk assessment (TARA) approach during the design phase of the TCU running on a Morello Board. Although it is based on the recommended process from ISO/SAE 21434 [10], our approach is tailored to suit the design process of TCUs (at both system and sub-system levels) and the collaboration with design engineers. In particular, our approach has two main steps, as depicted in Figure 3. In the first step, system functions are defined based on use cases for the TCU as the initial input. Each use case comprises a set of related features and each feature is an abstract description of how the system performs to fulfil it. The function definition process provides a high-level system design for realising the feature. This design includes a system architecture capturing the main components and how they are connected, a set of functional requirements to realise the feature and a set of sub-system functional requirements further refining the functional requirements provided by the TCU. The second step starts with the design provided by the previous one. First, the description of the functional and subsystem functional requirements allows us to identify assets from the system by extracting data flow between system/subsystem components. Damage scenarios are then determined and rated by assuming when one of the security properties (confidentiality, integrity, and availability) is violated. The process continues with a STRIDE-based enumeration of all potential threats to the system. These threats are associated with relevant assets and hence damage scenarios. Each of them is further refined by one or more attack paths with their corresponding feasibility rated accordingly. We validate the attack paths from an external point of view by first reviewing the system architecture and, then, establishing relevant attack objectives and their realisation that are formulated in the forms of attack trees. Finally, the risk of each threat is assessed based on the associated impact and feasibility ratings.

In the remainder of this section, we introduce the set of tools, highlighted by reddish rectangles in Figure 3, which are employed in our approach. They include SysML for defining functions, rating for impact and feasibility, and the risk matrix for risk assessment. Finally, we provide the design of the use case “OTA software update”. It will be the input for the TARA process which will be discussed in more detail in the next section.

SysML

SysML (Systems Modelling Language) is a graphical modelling language used for specifying, analysing, designing and verifying complex systems. It provides a high-level abstraction of the system and its behaviour, enabling engineers to specify the overall structure and functionality of the system. In particular, engineers can use structure diagrams in SysML to model system architectures and sequence diagrams to model their behaviours. In the automotive industry, SysML is used to define the functions and behaviour of various components and systems within a vehicle. By using SysML to model vehicle systems and use cases, engineers can ensure that the vehicle is designed to meet its functional requirements, in terms of safety and cybersecurity alike.

STRIDE

The STRIDE method is a software-centric threat classification methodology, originally developed by Microsoft. The method allows threat identification in the design phase of any software or hardware and as such gives insight into potential attack scenarios. The threats are divided into six different categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. STRIDE model extends the original CIA model by correlating threats with security attributes: authentication, integrity, non-repudiation, confidentiality, availability, and authorisation.

Impact Rating

The impact of damage scenarios is rated by that on safety, financial, operational, and privacy as suggested by [10]. They are utilised to assess the impact on the road user, who is considered the primary stakeholder.

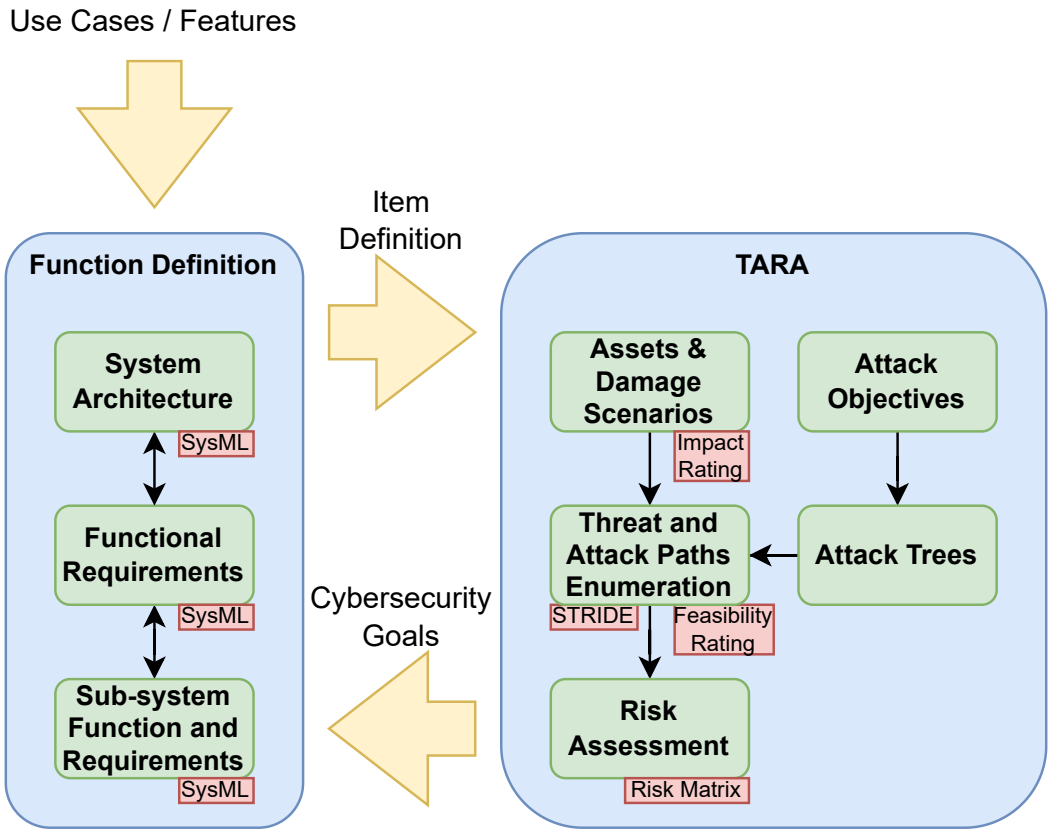

FIGURE 3 Our DSbD methodology to address cybersecurity risks for automotive embedded systems.

The **safety impact rating criteria**, taken from [8], classify the level of safety consequences. A severe rating represents deadly injuries with uncertain survival, while a major rating indicates deadly injuries with possible survival. A moderate rating signifies light injuries and a negligible rating implies no injuries.

The **financial impact rating criteria** assess the level of financial consequences. The severe rating signifies high financial damage that the stakeholder may not be able to overcome. Major rating represents significant financial damage that the stakeholder will be able to overcome. Moderate rating means that it will cause inconvenient results, but the stakeholder will be able to overcome with limited sources. A negligible rating implies no financial damage, and the stakeholder will not take any action.

In terms of **operational impact rating,** severe rating indicates that the vehicle becomes non-operational. The major rating signifies the loss of vehicle function. A moderate rating implies partial dysfunction or performance loss of the vehicle. A negligible rating means there is no effect on the vehicle's function or performance from the damage.

The **privacy impact rating criteria** classify the level of impact on the road user and the sensitivity of the information involved. The severe rating signifies an irreversible impact on the road user, with highly sensitive information that can easily be linked to the Personally Identifiable Information (PII) principal [7]. A major rating indicates a serious effect on the road user, with highly sensitive information either difficult to link to PII or sensitive and easy to link to PII. A moderate rating signifies inconvenience caused to the road user, with information that is sensitive but challenging to link to PII or non-sensitive but easy to link to PII. A negligible rating implies no effect on the road user, with information that is not sensitive and difficult to link to PII.

Attack Feasibility Rating

The feasibility of an attack indicates the ease or difficulty of carrying out an attack on four distinct levels. A high feasibility level suggests that the attack path is easy to accomplish. A medium feasibility level implies that the attack path is feasible and commonly encountered. A low feasibility level indicates that the attack path is feasible to some extent. Finally, a very low feasibility level suggests that it is highly challenging, if not nearly impossible, to accomplish the attack path.

In our work, the attack feasibility is determined by the attack potential-based rating approach [9]. This is one of the three approaches suggested in [10]. In this approach, one has to take into account different aspects of the attack including elapsed time, specialist expertise, knowledge of the item (or component), window opportunity, and equipment.

The **elapsed time** scale measures the duration required to discover a vulnerability, create an exploit, and successfully execute it. The rating assigned on this scale is influenced by the prevailing level of expertise and knowledge at the time of the assessment. There are values defined for each parameter based on [9] as depicted in Table 1.

Elapsed Time	
Enumerate	Value
<1 week	0
<1 month	1
<6 months	4
<= 3 years	10
>3 years	19

TABLE 1 Rating levels and corresponding values for Elapsed Time.

The **expertise** of an attacker plays a crucial role in their capabilities. It is categorized into four levels. Layman refers to individuals with no particular knowledge or expertise. Proficient attackers possess knowledge of security behaviour. Experts are familiar with the underlying algorithms and concepts specific to the targeted product. In some cases, multiple experts from different fields collaborate when their combined expertise is required. There are values defined for each parameter based on [9], as depicted in Table 2.

Specialist Expertise	
Enumerate	Value
Layman	0
Proficient	3
Expert	6
Multiple experts	8

TABLE 2 Rating levels and corresponding values for Elapsed Time.

The **knowledge of the item or the component** an attacker obtains is divided into four categories. Public information is readily available through the Internet. Restricted information is shared by the developer organization with select parties. Confidential information is limited to discrete teams within the same organization, confidential information is known only to a few people, and access is strictly controlled. There are values defined for each parameter based on [9], as depcited in Table 3.

Knowledge of the Item (or Component)	
Enumerate	Value
Public	0
Restricted	3
Confidential	7
Strictly Confidential	11

TABLE 3 Rating levels and corresponding values for knowledge of the item.

The **window of opportunity** refers to the likelihood of a successful attack. An unlimited window implies access available remotely from public networks without time limitations. Easy access allows remote entry for a limited time. Moderate access suggests limited physical access to the target without requiring specific tools. Difficult access indicates high-security measures, making it challenging to find an opportunity to launch an attack. There are values defined for each parameter based on [9], as depicted on Table 4.

The **equipment** used in an attack varies in accessibility and specificity. Standard equipment is readily available to the attacker and may even be a part of the targeted item itself. Specialized equipment is not initially accessible but can be easily obtained

Window of Opportunity	
Enumerate	Value
Unlimited	0
Easy	1
Moderate	4
Difficult/None	10

TABLE 4 Rating levels and corresponding values for knowledge of the item.

from public sources. Bespoke equipment is not accessible to the attacker and cannot be obtained publicly. It often requires custom development and can be costly. Multiple bespoke equipment refers to the use of custom-built tools for specific stages of the attack. There are values defined for each parameter based on [9], as depicted in Table 5.

Equipment	
Enumerate	Value
Standard	0
Specialized	4
Bespoke	7
Multiple bespoke	9

TABLE 5 Rating levels and corresponding values for equipment.

Attack potential mapping: Once each feasibility aspect of the attack is determined, attack potential is defined by the addition of numerical values of all the parameters that are shown in Tables 1 to 5. After each value from one parameter is summed up, the final numerical value and the corresponding attack feasibility rating can be looked up in Table 6.

Values	Attack Feasibility
0-9	High
10-13	High
14-19	Medium
20-24	Low
=>25	Very low

TABLE 6 The corresponding between attack feasibility values and ratings.

Risk Matrix

Once the damage impact and attack feasibility ratings are determined for an attack, its risk can be looked up by using a risk matrix. It is a visual tool that levels of impact and attack feasibility to determine risk values. It serves multiple purposes, including supporting decision-making for risk treatment and control selection, prioritising risks, reporting to stakeholders, and monitoring risk. Organisations have the flexibility to customise risk matrices according to their specific requirements, such as using different matrices for different types of damage.

In this approach, we follow IEC 31010 for determining risk values, using a scale of 1 to 5 where 1 is minimal risk and 5 is the highest risk. The mapping from impact and attack feasibility to specific risk values is specified in Figure 7.

Risk Matrix		Attack Feasibility			
		Very low	Low	Medium	High
Impact	Severe	1	3	4	5
	Major	1	2	3	4
	Moderate	1	2	2	3
	Negligible	1	1	1	1

TABLE 7 Risk matrix used to determine the risk value based on the values of impact ratings and attack feasibility ratings.

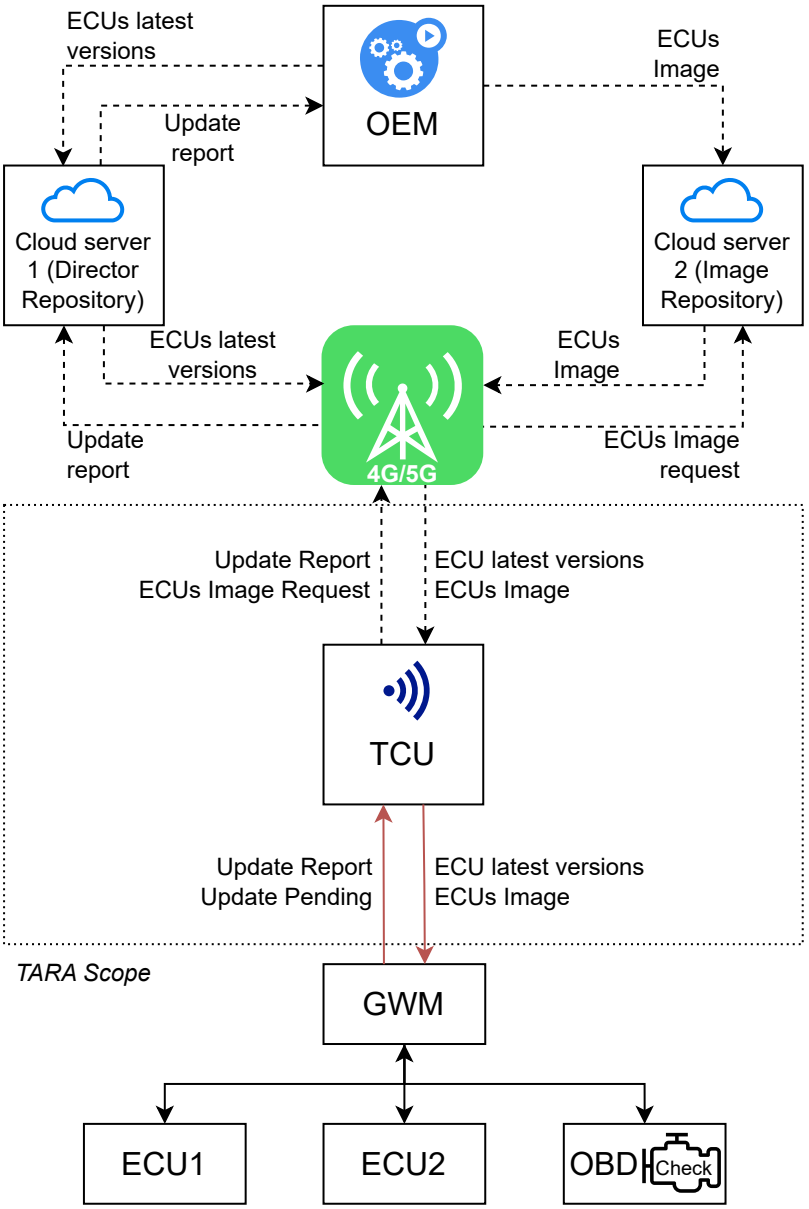

FIGURE 4 The system-level architecture of the OTA Firmware Updates system. Wireless communications are denoted by - - - - - - -, Ethernet by ———, and CAN by ———.

OTA Software Update Design

The system implements an over-the-air update process similar to that recommended in the Uptane specifications [19], as briefly recalled in Section 2. At a high level, the system covers the building of the software update, delivery to the TCU, and application of the update to one or more vehicle ECUs.

The structure of the OTA Firmware Updates system is designed at a system level and a sub-system level. At the system level, as depicted in Figure 4, it consists of the following several key components:

- GWM (Gateway Module) acts as the central communication hub within the vehicle, and the ECUs update OTA Manager. It facilities communication between various components such as ECUs and TCU and coordinates the firmware update process.
- ECUs (Electronic Control Units) are responsible for controlling specific vehicle functions. The GWM communicates with ECUs to coordinate the firmware update process and distribute the updates to the relevant ECUs.

- TCU (Telematics Control Unit) serves as the connection point between the vehicle and the cloud servers. It leverages 4G/5G communication capabilities to establish a secure and reliable connection for OTA updates.
- 4G/5G Communication - network infrastructure provides the connectivity required for the TCU to communicate with the cloud servers/
- Cloud Server 1 Director Repository is responsible for managing update campaigns and defining update policies.
- Cloud Server 2 Image Repository stores the firmware images and associated metadata. The TCU communicates with this repository to retrieve the required firmware images for the vehicle's ECUs.
- OEM plays a crucial role in managing the overall OTA firmware update process, such as defining update campaigns, managing Director Repository and ensuring the integrity and security of the firmware updates.

These components must operate and interact with each other to deliver the OTA Firmware Update functionality. Table 8 lists all the system-level functional requirements which are elicited for these components of the OTA Firmware Updates system in order to enable over-the-air update functionality for vehicles. In this design, it is imperative that there is a trust established

ID	Description
F1	TCU configures IVI ECU versions reading strategy
F2	TCU reads/stores ECUs latest version from Cloud server 1
F3	TCU provides ECUs latest versions to GWM
F4	TCU reads update pending from GWM
F5	TCU requests ECU image repository to update from Cloud server 2
F6	TCU provides ECU Image repository to the GWM
F7	TCU reads update report from the GWM
F8	TCU provides update report to OEM
F9	TCU establishes secured GWM communication between its CAN bus communication and GWM
F10	TCU establishes secured communication between its 4G/5G module with the cloud server 1
F11	TCU establishes secured communication between its 4G/5G module with the cloud server 2

TABLE 8 List of functional requirements for the OTA Software Update Use Case.

between the ECUs and the software build and signing system to ensure that only verified updates are downloaded and installed. In addition to the integrity of the update, the confidentiality of the interaction payloads needs to be guaranteed to protect proprietary intellectual property and reduce the likelihood of vulnerability discovery through reverse engineering.

The design also goes one level deeper to identify the system architecture at the sub-system level, as depicted in Figure 5. At this level, our engineers look at the sub-components of the TCU (including the microcontroller (which also contains a Real-Time Controller (RTC)), memories and communication modules (4G/5G, GPS, Ethernet, CAN)) to elicit additional sub-system functional requirements. For example, let us consider functional requirement 5 (TCU requests ECU image repository to update from Cloud server 2) as listed in Table 8. This requirement is broken down into 6 sub-system functional requirements (1) to prepare the request, (2) to establish the communication, (3) to send the request, (4) to receive the data, (5)to read the ECU image from the data, and (6) to store the received ECU image into the volatile memory (VM). These sus-system functional requirements are listed in Table 9.

ID	Description
SF27	TCU prepares ECU image repository request to cloud server 2
SF28	TCU establishes cloud server 2 secured communication
SF29	TCU sends ECU image repository request to cloud server 2
SF22	TCU reads received Cloud server 2 data
SF30	TCU reads ECU image repository from Cloud server 2
SF51	TCU stores ECU image repository packages in VM

TABLE 9 List of sub-system functional requirements for F5.

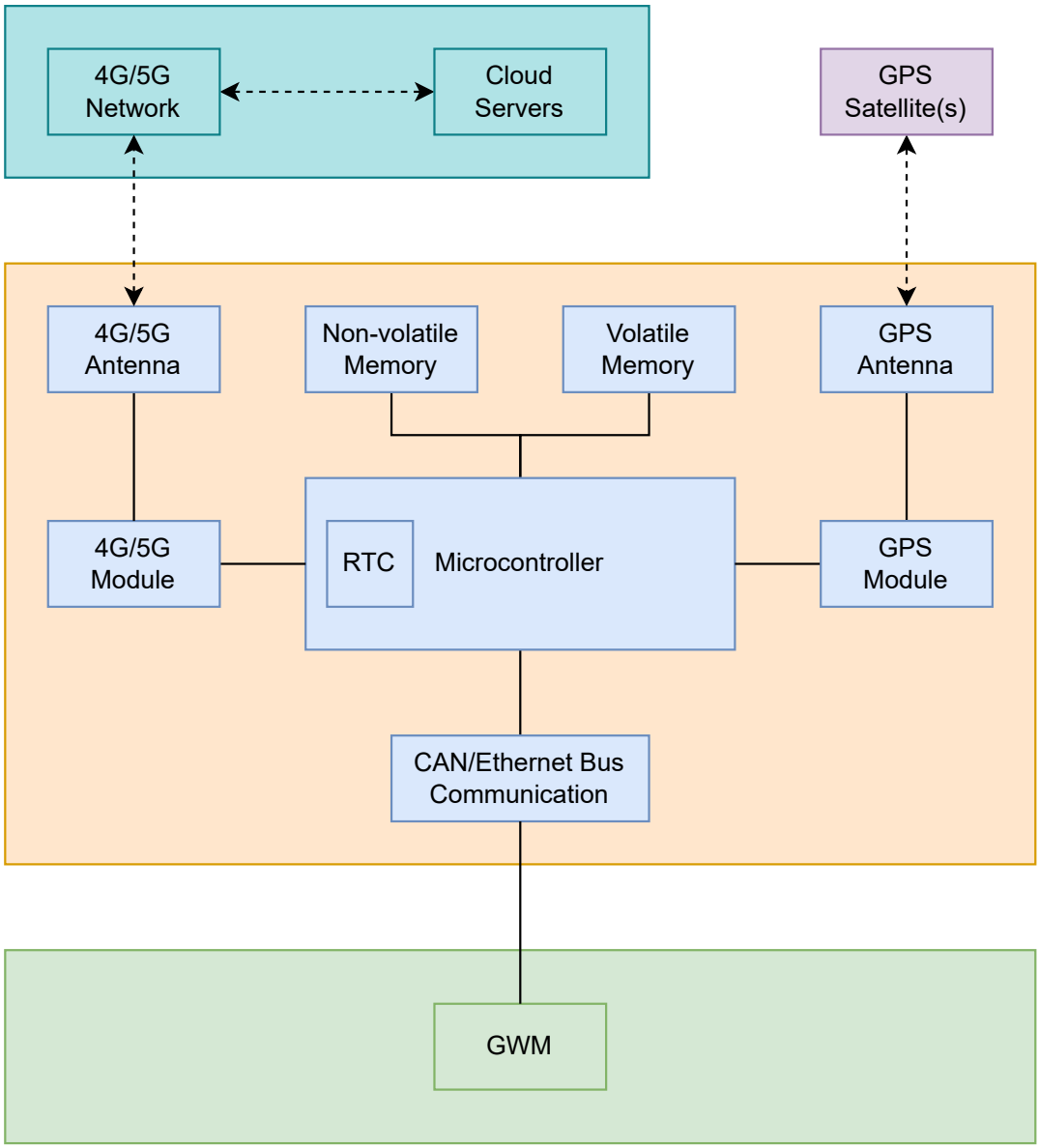

FIGURE 5 The sub-system level architecture.

4 | TARA RESULT

In this section, we summarise the result of the TARA process applied to the design of the OTA Firmware Update system. Both the process and the design are described in Section 3.

4.1 | Identifying Assets and Damage Scenarios

Following the methodology discussed in Section 2, the TARA process is begun by identifying assets. The identification is based on the input from the design including the structure diagrams and functional requirements. At the system level, relevant assets are functional input/output data transferred between different components of the system. At the sub-system level, relevant assets are input/output data for each sub-system function. To this end, we have identified 29 assets at the system level and 60 at the sub-system level.

ID	Damage Scenario	Impact Rating				
		S	F	O	P	Overall
D01	Software update is blocked	4	1	3	1	4
D02	Disclosure of ECU latest version	1	1	1	3	3
D03	Software update is rolled back to an old version	4	1	3	1	4
D04	Disclosure of ECU firmware	1	3	1	3	3
D05	Malicious firmware is installed	4	1	3	1	4

TABLE 10 List of damage scenarios and their rating for the OTA Software Update Use Case. Impact rating criteria: S for Safety, F for Financial, O for Operational and P for Privacy. Impact rating levels: 4 for severe, 3 for major, 2 for moderate and 1 for negligible.

Each asset is analysed to determine if any security properties of confidentiality, integrity, and availability may be compromised, which could lead to various damage scenarios. Ultimately, we have identified a total of 5 potential damage scenarios as listed in Table 10.

In the following, we illustrate this process of asset and damage identification in detail for the case of a system functional requirement F5 (TCU requests ECU image repository to update from the Cloud server 2) and its sub-system functional requirement SF30 (TCU reads ECU image repository from Cloud server 2) as listed in Table 9. Assets and their associated damage scenarios for other requirements can be found in the complete TARA result[‡].

At the system level, an asset, with ID A21, associated with F5 is identified. It is the request for up-to-date ECU images that sent from the TCU to the Cloud server 2. A violation of A21's confidentiality will lead to the disclosure of the ECU latest version in the request. This is damage scenario D02 where the attacker can monitor the latest version of the ECU firmware. Alternatively, the attack might want to modify the request, i.e., violating the integrity of A21. If he changes the version number in the request to the current one, the corresponding ECU will update its firmware to the current version that it already has. This is equivalent to no update to the actual latest version. This corresponds to damage scenario D01 where the actual software update is blocked. If the attacker changes the version number in the request to an older one, the ECU will eventually install firmware that is older than the one it currently has. This is damage scenario D03 where the software update is rolled back to an old version. Finally, the attack can attempt to block the request from the TCU to reach the Cloud server 2. This attack violates the availability of A21 and results in damage scenario D01 which blocks software updates. These damage scenarios associated with A21 are summarised in Table 11.

Asset	**C/I/A**	**Damage Scenario**	**Overall Impact**
A21	C	D02	Major
A21	I	D01	Severe
A21	I	D03	Severe
A21	A	D01	Severe

TABLE 11 Damage scenarios for A21 (the request of ECU image from TCU to Cloud Sever 2).

At the sub-system level, two assets associated with SF30 are identified. The first asset, with ID A62, is the implementation of SF30 running in the microcontroller of the TCU. The second asset, with ID A63, is the ECU image stored in VM. It is the output of SF30. Let us consider asset A63, the violation of A63's confidentiality will lead to the disclosure of the ECU firmware. This is damage scenario D04 where an attack will collect the ECU firmware, potentially for further analysis. The violation of A63's integrity can lead to either the rolling back of the software update to a previous version (damage scenario D03) or the installation of malicious firmware (damage scenario D05). In the first case, the attacker replaces the ECU image with a previous version which could mean bugs in the previous version are reintroduced which could lead to safety and/or operational issues. If it is related to the brake system, unpatched bugs could lead to accidents and loss of life. In the second case, the attacker even introduces the ECU image with malicious behaviours. For example, malicious control of the brake system can lead to accidents and loss of life. Therefore, safety is rated as severe for both of these damage scenarios. Finally, the violation of A63's availability can lead to the blocking of software updates (damage scenario D01). In this case, the attack prevents access to the ECU image stored in VM. Hence, it cannot be forwarded via GWM to the relevant ECU for updating. If the firmware of the

[‡] The complete TARA result can be downloaded from https://tinyurl.com/autocheritara.

ECU contains a bug, this could lead to accidents and loss of life. This is why the safety rating of D01 is severe. These damage scenarios associated with A63 are summarised in Table 12.

Asset	C/I/A	Damage Scenario	Overall Impact
A63	C	D04	Major
A63	I	D03	Severe
A63	I	D05	Severe
A63	A	D01	Severe

TABLE 12 Damage scenarios for A63 (ECU image stored in VM).

4.2 | Threat and Attack Path Analysis

Potential threats to the OTA Software Update use case are enumerated by applying STRIDE (see Section 3). This means threats are categorised into six groups: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege. To provide a comprehensive list of threats for the considered use case, we consider all attack surfaces at the system and sub-system levels. At the system level, the considered attack surfaces are the wireless communication between the TCU and the Cloud servers and the CAN bus communication between the TCU and the GWM. Threats are further enumerated for each of these attack surfaces based on the transmission direction. At the sub-system levels, the considered attack surfaces are the microcontroller (MC), the volatile memory (VM) and the non-volatile memory (NVM). We then perform a high-level attack path analysis for each enumerated threat. This includes attack steps that will be carried out to realise the threat, starting from compromising an attack surface and finishing with achieving the goal of the threat. In total, we found 33 threats and 43 attack paths in the OTA software update use case.

In the following, we provide a more detailed analysis of the two assets A21 (ECU image request) and A63 (ECU image stored in the volatile memory). The analysis for other assets can be found in the complete TARA result. At the system level, let us consider asset A21 (ECU image request) that is sent from the TCU to the Cloud server 2. Threats to this asset are enumerated for the wireless communication attack surface where the direction of the data transmission is from the TCU to the Cloud servers. Once the wireless communication between the TCU and the cloud servers is compromised, the following threats are identified:

- T07 - Spoofing: Spoofed messages are sent from the TCU to the cloud servers;
- T08 - Tampering: Messages sent to cloud servers are manipulated;
- T10 - Information Disclosure: Messages sent to cloud servers are captured and recorded;
- T11 - Denial of Service: Messages sent to cloud servers are blocked;
- T12 - Elevation of privilege: TCU is compromised and sends malicious messages to the cloud servers.

Threats T07, T08 and T12 violate the integrity of A21. This means they can lead to damage scenarios D01 and D03. Hence, their overall impact is the highest impact of D01 and D03, i.e., severe. Threat T10 violates the confidentiality of A21. Therefore, it can lead to damage scenario D02. Hence, its overall impact is major. Finally, threat T11 violates the availability of A21. Therefore, it can lead to damage scenario D01. Hence, its overall impact is severe. These threats and their corresponding impacts are summarised in Table 13. The attack path analysis for these threats is as follows. We identify one attack path to realise threat

Threat	STRIDE	Overall Impact
T07	Spoofing	Severe
T08	Tampering	Severe
T10	Information Disclosure	Major
T11	Denial of Service	Severe
T12	Elevation of Privilege	Severe

TABLE 13 Enumerating STRIDE threats for wireless communication and A21 and their associated impacts.

T07. In this attack path, denoted by AP06, an attacker must first connect to the wireless network between the TCU and the cloud

servers. The attack then impersonates the TCU and sends spoofed messages to the cloud servers. To analyse the feasibility of the attack path, we estimate that the attacker must be an expert so that he knows the employed wireless technology and the authentication mechanism used between the TCU and the cloud server. He must also possess the secret shared between the TCU and the cloud server to enable authentication, i.e., the knowledge about the OTA software update system is at the confidential level. The window opportunity of the attack is moderate as he might need physical access to the TCU to gain the shared secret. He also needs to use specialised equipment to create rough wireless access points. To this end, we estimate that it must take him about a month to complete the attack. Overall, the attack potential for AP06 is 22 and, therefore, the corresponding attack feasibility rating for AP06 is low. Similarly, threats T08 and T09 can be realised by first connecting to the wireless network and then poisoning the network so that all traffics are diverted through the attacker's device. From the, the attacker can either manipulate or eavesdrop on the messages to be sent to the cloud servers. T11 can be realised by two different attack paths. One is to perform a DoS attack against the TCU and the other is to disrupt the wireless communication by a jamming device. Finally, T12 can be realised by injecting malicious software to the TCU which will sending out spoofed messages to the cloud servers. Their attack feasibility ratings of these attack paths are summarised in Table 14.

Threat	APID	Attack Path	AF
T07	AP06	Connect to Wireless Network → Impersonate TCU → Send spoofed message to the cloud servers	Low
T08	AP07	Connect to Wireless Network → Poison the network to direct all traffic through attacker's device → Manipulate messages from TCU to the cloud servers	Very Low
T10	AP08	Connect to Wireless Network → Poison the network to direct all traffic through attacker's device → Eavesdrop messages from TCU to the cloud servers	Medium
T11	AP09	Connect to Wireless Network → Perform a DoS attack against TCU	Medium
	AP10	Use a jamming device to disrupt the Wireless Communication	Very Low
T12	AP11	Connect to Wireless Network → Compromise the TCU by injecting malicious software or replacing the firmware containing malicious software via unsecured OTA → Spoofed messages are sent by the malicious software	Very Low

TABLE 14 Attack path analysis for threats T07, T08, T10, T11 and T12. APID: Attack Path ID. AF: Attack Feasibility rating.

At the sub-system level, let us consider asset A63 (ECU image stored in the volatile memory). Threats to this asset are enumerated for the volatile memory. Once the volatile memory is compromised, the following threats are identified:

- T31 - Spoofing: Data is spoofed in the volatile memory;
- T32 - Tampering: Data in the volatile memory is manipulated;
- T34 - Information Disclosure: Data in the volatile memory is leaked out;
- T35 - Denial of Service: Access to data in the volatile memory is blocked;
- T36 - Elevation of privilege: TCU is compromised and the volatile memory is controlled by malicious software.

Note that all of these threats are specific to CHERI as they can be defined by the memory protection mechanism designed by CHERI. Similar to the threats on A21 and their impacts, we repeat the same STRIDE and attack path analyses for A63. The impacts of the above threats on A63 are summarised in Table 15. To realise T31, the attacker first gains access to the volatile

Threat	STRIDE	Overall Impact
T31	Spoofing	Severe
T32	Tampering	Severe
T34	Information Disclosure	Major
T35	Denial of Service	Severe
T36	Elevation of Privilege	Severe

TABLE 15 Enumerating STRIDE threats for the volatile memory and A63 and their associated impacts.

memory and then fills it with spoofed data. This is identified as attack path AP27. Under the protection provided by CHERI, this attack likely requires the attacker to have multiple expertise such as both in embedded hardware and software. He might be required to have shared secrete about the encoding of the firmware on the embedded hardware. The window opportunity to access the volatile memory is also low as it is usually embedded within a SoC which requires multiple bespoke devices to access. Overall, the attack potential for this attack path is 35, which corresponds to the attack feasibility rating of very low. Similarly, T32 is realised by attack path AP28 where the attacker needs to gain access to the volatile memory and manipulate its content. T34 and T35 are realised by compromising the volatile memory, and then either (AP29) sending out the data on the volatile memory or (AP30) blocking access to the volatile memory. Finally, T36 is obtained by (AP31) compromising the TCU and then taking control of the volatile memory by malicious firmware. Their attack feasibility ratings of these attack paths are summarised in Table 16.

Threat	APID	Attack Path	AF
T31	AP27	Gain access to the volatile memory → Fill it with spoofed data	Very Low
T32	AP28	Gain access to the volatile memory → Manipulate its data	Very Low
T34	AP29	Compromise the volatile memory → Send its data to attacker	Very Low
T35	AP30	Compromise the volatile memory → Block access to the volatile memory	Very Low
T36	AP31	Compromise the TC → Take full control of the volatile memory	Very Low

TABLE 16 Attack path analysis for threats T31, T32, T34, T35 and T36. APID: Attack Path ID. AF: Attack Feasibility rating.

4.3 | Risk assessment result

Once the attack path analysis is complete, we use the risk matrix in Table 7 to evaluate the risk value for each of the 43 identified attack paths. In particular, 26 attack paths are rated to have the lowest risk value of 1, 13 attack paths are rated to have a risk value of 3 and 4 attack paths are rated to have a risk value of 4. None of the attack paths are rated to have the highest risk value of 5. All 4 attack paths rated 4 are related to denial of service threats which prevent the OTA software update. Thanks to the utilisation of CHERI and the Morello Board, all attack paths related to threats against the volatile memory have the lowest risk value of 1.

In the following, we provide the detail for attack paths related to assets A21 (ECU image request) and A63 (ECU image stored in the volatile memory). As presented in Table 14, A21 has 6 associated attack paths. Their risk assessment is summarised in Table 17. The highest risk (value of 4) is associated with attack path AP09 which performs a DoS attack against the TCU to prevent the OTA software update from being carried out. This violates the availability of A21, which leads to a severe impact on safety. Similarly, A63 has 5 attack paths against the volatile memory as listed in Table 16. Their risk assessment is summarised in Table 18 where all have the lowest risk value of 1.

Threat	APID	Impact	AF	Risk value
T07	AP06	Severe	Low	3
T08	AP07	Severe	Very Low	1
T10	AP08	Major	Medium	3
T11	AP09	Severe	Medium	4
	AP10	Severe	Very Low	1
T12	AP11	Severe	Very Low	1

TABLE 17 Risk values for attack paths associated with A63 (ECU image stored in the volatile memory).

Threat	APID	Impact	AF	Risk value
T31	AP27	Severe	Very Low	1
T32	AP28	Severe	Very Low	1
T34	AP29	Major	Very Low	1
T35	AP30	Severe	Very Low	1
T36	AP31	Severe	Very Low	1

TABLE 18 Risk values for attack paths associated with A63 (ECU image stored in the volatile memory).

4.4 | Security requirements from TARA

Based on the TARA result, we have proposed a set of security requirements to mitigate the identified risks. They can be categorised into two groups. The first group of security requirements is about securing communication with the TCU against threats such as those listed in Table 13 related to A21 (ECU image request). The second group is about securing the firmware of the TCU on the Morello board to defend against memory attack threats such as those listed in Table 15 related to A63 (ECU image stored in the volatile memory). These security requirements are listed below.

- Security requirement to secure TCU communication

  · The wireless communication between the Cloud Server and TCU shall follow a secure protocol.
  · Communication between the TCU and the Cloud Server/OEM shall be encrypted using a strong encryption algorithm.
  · A resource exhaustion detection system shall be used to monitor the vehicle system.

- Security requirements to secure the firmware of the TCU:

  · There shall be a secure boot process to prevent any invalid firmware or malicious code from being booted.
  · The TCU shall verify the legitimacy and authenticity of all data received from external sources.
  · Plausibility checks used for status information shall ensure authenticity, integrity and freshness. For example, live counters or checksums shall be used.
  · The TCU shall implement secure over-the-air (OTA) software update mechanisms that verify the authenticity of GWM firmware updates before installation.
  · Firmware update shall be signed and verified
  · Secrets and credentials shall not be stored in plaintext.

5 | CONCLUSION AND FUTURE WORK

Undoubtedly, the adoption of novel secure hardware by system designers and end-users depends on the clear value proposition articulated for the benefit of everyone involved in the supply chain [18]. Our attempt in this paper to present a (ISO21434) standards-driven threat analysis and risk assessment for the design of TCUs is to demonstrate the risk mitigation offered by CHERI-based hardware to the automotive ecosystem of suppliers and manufacturers. The CHERI-based hardware platform, namely the Morello Board, provides protection against memory attacks such as buffer overflow. The utilisation of the threat analysis and risk assessment is a part of the application of Secure-by-Design to the design process of the TCU. The result of the threat analysis and risk assessment has provided

- a deep and comprehensive understanding of potential risks that the TCU may have, and
- input for the formulation of necessary security requirements to ensure that these risks are properly dealt with.

The risks identified in our effort arise out of attacks that target the TCU memory and others that do not. The result has shown that the application of the Morello Board has a significant impact on reducing the risks of relevant attacks to memories, similar to the suggestion in [11]. Based on this result, we have also suggested a list of security requirements which can be subsequently considered in further steps from design and implementation to testing of TCUs. As such, this work contributes significantly

towards effective value for automotive cybersecurity controls, in a timely fashion given the imperatives around regulatory compliance.

In the future, we plan to extend our approach in terms of automation and investigate objective reasoning methods for evaluating impact and attack feasibility ratings. In our experiment, many activities in the threat assessment and risk analysis are tedious and repetitive. Therefore, an automation tool will have the potential to avoid mistakes, speed up the process, and reduce the involvement of cybersecurity experts. However, to reduce this involvement, it is vital to seek objective reasoning methods for accurately evaluating ratings for threats and attacks.

ACKNOWLEDGEMENT

This research was funded by Innovate UK (Project Number 10018347).

REFERENCES

[1] ARM. Arm morello program. https://www.arm.com/architecture/cpu/morello, 2023.

[2] CHERI. Capability hardware enhanced risc instructions (cheri). https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/, 2023.

[3] CISA. Security-by-design and -default. https://www.cisa.gov/resources-tools/resources/secure-by-design-and-default, 2023.

[4] P. Efstathiadis, A. Karanika, N. Chouliaras, L. Maglaras, and I. Kantzavelou. Smart cars and over-the-air updates. In *Cybersecurity Issues in Emerging Technologies*, pages 137–152. CRC press, Boca Raton, 2021.

[5] S. Halder, A. Ghosal, and M. Conti. Secure over-the-air software updates in connected vehicles: A survey. *Computer Networks*, 178:107343, 2020.

[6] C. Hammerschmidt. Number of automotive ecus continues to rise. https://www.eenewseurope. com/en/number-of-automotive-ecus-continues-to-rise/, 2019.

[7] ISO/SAE. ISO/IEC 29100:2011 Information technology — Security techniques — Privacy framework. Standard, International Organization of Standardization, 2011.

[8] ISO/SAE. ISO 26262-3:2018 Road vehicles — Functional safety — Part 3: Concept phase. Standard, International Organization of Standardization, 2018.

[9] ISO/SAE. ISO/IEC 18045:2022 Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Methodology for IT security evaluation. Standard, International Organization of Standardization, 2021.

[10] ISO/SAE. ISO/SAE FDIS 21434, Road Vehicles - Cybersecurity engineering. Standard, International Organization of Standardization, 2021.

[11] N. Joly, S. ElSherei, and S. Amar. Security analysis of cheri isa. https://github.com/microsoft/MSRC-Security-Research/blob/master/papers/2020/Security

[12] T. K. Kuppusamy, A. Brown, S. Awwad, D. McCoy, R. Bielawski, C. Mott, S. Lauzon, A. Weimerskirch, and J. Cappos. Uptane: Securing software updates for automobiles. In *14th ESCAR Europe*, 2016.

[13] S. Mahmud, S. Shanker, and I. Hossain. Secure software upload in an intelligent vehicle via wireless communication links. In *IEEE Proceedings. Intelligent Vehicles Symposium*, pages 588–593, 2005. .

[14] A. T. Markettos, J. Baldwin, R. Bukin, P. G. Neumann, S. W. Moore, and R. N. M. Watson. Position paper:defending direct memory access with cheri capabilities. In *Hardware and Architectural Support for Security and Privacy*, HASP '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450388986. . URL https://doi.org/10.1145/3458903.3458910.

[15] R. Messnarz, D. Ekert, G. Macher, A. Much, T. Zehetner, and L. Aschbacher. Experiences with the automotive spice for cybersecurity assessment model and tools. *Journal of Software: Evolution and Process*, 35(9):e2519, 2023. . URL https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2519.

[16] D. K. Nilsson and U. E. Larson. Secure firmware updates over the air in intelligent vehicles. In *ICC Workshops - 2008 IEEE International Conference on Communications Workshops*, pages 380–384, 2008. .

[17] M. Steger, A. Dorri, S. S. Kanhere, K. Römer, R. Jurdak, and M. Karner. Secure wireless automotive software updates using blockchains: A proof of concept. In C. Zachäus, B. Müller, and G. Meyer, editors, *Advanced Microsystems for Automotive Applications 2017*, pages 137–149, Cham, 2018. Springer International Publishing. ISBN 978-3-319-66972-4.

[18] A. Tomlinson, S. Parkin, and S. A. Shaikh. Drivers and barriers for secure hardware adoption across ecosystem stakeholders. *Journal of Cybersecurity*, 8(1):1–14, 08 2022. ISSN 2057-2085. . URL https://doi.org/10.1093/cybsec/tyac009.

[19] Uptane Alliance. Ieee-isto 6100.1.0.0 uptane standard for design and implementation, 2019. https://uptane.github.io/papers/ieee-isto-6100.1.0.0.uptane-standard.html, Last accessed on 2019-12-6.

[20] M. Zahid, I. Inayat, M. Daneva, and Z. Mehmood. Security risks in cyber physical systems—a systematic mapping study. *Journal of Software: Evolution and Process*, 33(9):e2346, 2021. . URL https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2346.