Hindawi
Security and Communication Networks
Volume 2019, Article ID 8348925, 14 pages
https://doi.org/10.1155/2019/8348925

Research Article

Security Requirements Engineering in Safety-Critical Railway Signalling Networks

Markus Heinrich,[1] **Tsvetoslava Vateva-Gurova**,[1] **Tolga Arul**,[1] **Stefan Katzenbeisser**,[1] **Neeraj Suri**,[1] **Henk Birkholz**,[2] **Andreas Fuchs**,[2] **Christoph Krauß**,[2] **Maria Zhdanova**,[2] **Don Kuzhiyelil**,[3] **Sergey Tverdyshev**,[3] **and Christian Schlehuber**[4]

[1]*Department of Computer Science, TU Darmstadt, Germany*
[2]*Fraunhofer Institute for Secure Information Technology (SIT), Germany*
[3]*SYSGO AG, Germany*
[4]*DB Netz AG, Germany*

Correspondence should be addressed to Markus Heinrich; heinrich@seceng.informatik.tu-darmstadt.de

Received 21 December 2018; Revised 20 March 2019; Accepted 16 May 2019; Published 14 July 2019

Academic Editor: David Megias

Securing a safety-critical system is a challenging task, because safety requirements have to be considered alongside security controls. We report on our experience to develop a security architecture for railway signalling systems starting from the bare safety-critical system that requires protection. We use a threat-based approach to determine security risk acceptance criteria and derive security requirements. We discuss the executed process and make suggestions for improvements. Based on the security requirements, we develop a security architecture. The architecture is based on a hardware platform that provides the resources required for safety as well as security applications and is able to run these applications of mixed-criticality (safety-critical applications and other applications run on the same device). To achieve this, we apply the MILS approach, a separation-based high-assurance security architecture to simplify the safety case and security case of our approach. We describe the assurance requirements of the separation kernel subcomponent, which represents the key component of the MILS architecture. We further discuss the security measures of our architecture that are included to protect the safety-critical application from cyberattacks.

1. Introduction

The integration of commercial off-the-shelf (COTS) hardware and software into industrial control systems such as railway command and control systems (CCSs) is in progress. However, introducing COTS components into a previously proprietary safety system leads to novel security threats. The interplay between safety and security is an active research area, where many questions are yet to be answered. An extensive survey of approaches to combine safety and security in industrial control systems has been performed by Kriaa et al. [1]. Our study of the safety-security interplay is motivated by the lack of a security architecture for railway signalling.

Current train control is centralized in a signal box, also called interlocking system (ILS), that controls a defined area of the tracks comprising of multiple track switches (points) and signals. An example with a single point and signal is shown in Figure 1. If a train needs to move on the tracks, the ILS sets the points according to the desired route. If the movement of the train is considered safe, the ILS sets the signal for the route to clear. The aspect of the signal is observed by the train driver, who is allowed to safely proceed on the journey. A route is considered safe for a train, if it is not occupied by or reserved for another train, thus precluding collisions.

Points, signals, and other controllable objects are summarized under the term field elements. Earlier ILS generations used analogue signal transmission to set their field elements. Modern ILSs utilize IP networks to transmit their commands digitally to an object controller (OC) that in turn steers the

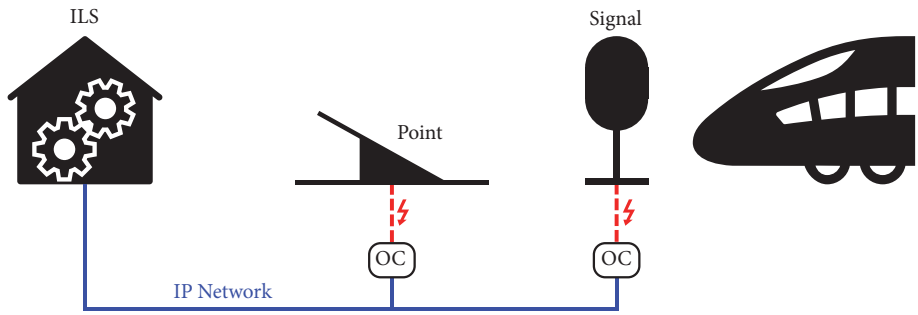

Figure 1: A railway command and control system.

field element by starting and stopping the point machine or turning on and off the signal's light bulbs, respectively. This allows for decoupling energy supply and command transmission and thus for larger distances between ILS and field element. Since railway signalling networks are classified as critical infrastructure (CI), the impact of potential security incidents on the railway system can be huge. This calls for the need of a security concept to ensure the robustness of railway signalling networks against cyberattacks. Furthermore, the railway system as a CI must meet national safety regulations. To address these issues, we execute a railway domain-specific requirements engineering process that has been proposed for German railways [2] but can be used as a template for international railway operation. The output of this process provides the foundation of a security architecture, which we propose for railway CCS.

Our contribution consists of the following. We report on our experience with the requirements engineering process of DIN VDE V 0831-104, and make suggestions for improvements. Then, we investigate the effect of these requirements on our case study, the safety-critical railway CCS. We show and discuss the derived security requirements for the case study. Subsequently, we use the identified threats and requirements to propose a security architecture for mixed-criticality systems such as the railway CCS running safety-critical applications along nonsafety-critical security applications. A mixed-criticality system must be carefully designed in order to maintain functionalities such as dependability and responsiveness under constrained resources and in the presence of attackers [3]. We propose the Secure Object Controller (SecOC), a security architecture based on a hardware platform that includes a hardware trust anchor. On top of the hardware, a separation kernel (SK) provides a software framework that allows running applications of mixed-criticality on our platform. On the software platform, safety and security applications coexist. The security applications protect the safety-critical application from cyberattacks. Complementary, the SK ensures that the security applications cannot exhaust the resources required by the safety application to fulfill its safety-critical task. Additionally, to enhance the security of the system, we apply security measures. An authenticated boot process uses the hardware trust anchor to ensure that only authorized software instances are executed on the hardware platform. A health monitor observes the system state during runtime and can report conspicuous state changes. A secure update mechanism allows for altering the system software, firmware, and configuration from authorized sources only. We further discuss our approach to handle assurance of the mixed-criticality system towards multiple certification authorities. Also, we show how the proposed architecture fulfills the requirements we derived.

The paper is organized as follows. Section 2 presents the system model. Section 3 details the conducted threat analysis. The elicited security requirements are presented in Section 4. Section 5 discusses the shortcomings of the conducted process as well as the suggested enhancements. Section 6 describes and discusses the proposed security architecture. Section 7 justifies that our security architecture fulfills the security requirements. Section 8 concludes the paper.

2. System Model

Our system model is a general railway signalling architecture composed of the interlocking layer and the field element layer, as depicted in Figure 2.

The interlocking layer encompasses the ILS and a maintenance and data management system (MDM). The MDM is responsible for updating the software of the components in the interlocking and field element layer as well as for time synchronization, and diagnostic data collection due to legal requirements. The heart of the signalling network is the ILS. It is responsible for issuing commands to the field elements of the lower architectural layer to execute the orders of the traffic supervisor, who is not considered in our system model. The ILS guarantees safe train operation by discarding unsafe orders, e.g., orders that would lead to the collision or derailment of trains.

On the lower layer, point machines are driven to switch tracks and clearances are communicated to train drivers via light signals. The commands to the OCs are sent through the network using specialized railway protocols. The OCs, located in junction boxes close to the tracks, switch the power of their corresponding field element (points, signals). There is a one-to-one relation between field element and OC. Thus, OCs are spatially distributed with limited physical protection by junction boxes and are therefore accessible by an attacker. In contrast, physical access to the components of the interlocking layer is more difficult to obtain, as

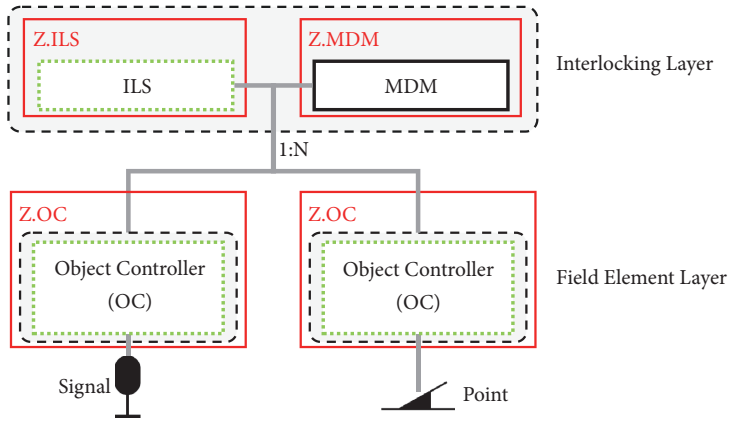

Figure 2: System model showing `Z.ILS`, `Z.MDM`, and two instances of `Z.OC`. Dashed frames indicate housing. Dotted frames indicate existing safety-critical functionality.

these components are subject to physical access control in a building.

The layers are connected via a network that is considered untrusted from a security perspective, because the railway operator has neither full control over its physical layout nor can guarantee its impenetrable protection. As a result, packets might be dropped, changed, rerouted, or read by an unknown third party.

The depicted building blocks are located in housings (indicated by the dashed line in Figure 2). This fact is important for the subsequent security analysis, because of the influence the housing can have on the attacker's capabilities to access the components. It protects the internal network of the interlocking layer and the enclosed objects such that certain malicious actions, e.g., physical penetration, become infeasible.

According to DIN VDE V 0831-104 [2], we split the system model into logical zones featuring components that are assumed to have equal security requirements. We identify three zones, named `Z.ILS`, `Z.MDM`, and `Z.OC`, shown in Figure 2. Our definition of zones includes the list of objects in the zone, the logical and physical borders, the data flows between the zones, the interfaces (Ethernet), and the physical connections of each zone. In our system definition, zones `Z.ILS` and `Z.MDM` reside in the same building and are instantiated only once per railway station. Many zones of type `Z.OC` exist in a single railway station and have weaker housing compared to `Z.ILS` and `Z.MDM`. The number of `Z.OC` is determined by the number of field elements controlled by a railway station and can reach up to 250 entities.

We use the system model consisting of the three zones and their interconnections as input for the next process step where we analyse the threats to the system.

3. Risk Analysis

To derive security requirements, we follow the German prestandard DIN VDE V 0831-104 for security in railway signalling networks [2]. It is designed as a guideline for applying the IEC 62443 framework for industrial control system security to railway signalling, while obeying to relevant railway safety standards such as EN 50128, EN 50129, and EN 50159.

In this section, we detail the conducted risk analysis, considering each of the zones of the system model (see Section 2) along with its components, their functions, the relevant data, and the communication between the components.

From a safety perspective, the risk is indicated as the product of the probability of an event to occur and the loss associated with the event. This approach has proven to be impractical for security analyses [4]. Attacks on security do not follow a probabilistic model of occurrence, because they are intentional. Moreover, even if security attacks could be characterized by a stochastic process, probabilities could not be easily determined due to the lack of a sufficient amount of samples and a constantly changing attacker landscape. Therefore, we conduct an explicit risk analysis, following one of the approaches proposed by DIN VDE V 0831-104, by using the presented system model.

3.1. Attacker Model. In order to describe capabilities of an attacker, DIN VDE V 0831-104 uses a set of qualitative dimensions: resource, knowledge, and motivation. The *resource* dimension reflects the financial and workforce capacity of the attacker to prepare and launch an attack. The *knowledge* dimension describes what she knows about the system before attacking it and can use to create opportunities for a successful offensive. These dimensions are characterized by numerical values from low (2) over medium (3) to high (4), where attackers with basic capabilities (1) are not considered [2]. Thus, *generic* knowledge (K = 2) comprises of publicly available information such as protocol specifications and COTS hardware. An attacker with *extended* knowledge (K = 4) has access to some closed information usually only available to a small circle of experts (insiders) working with the system. *Low* financial resources (R = 2) comprise a few thousand Euro while *extended* financial capacity (R = 4) provides resources in the magnitude of state actors.

Table 1: Examples of identified threats with assigned values for the attacker. FR: foundational requirement, R: resources, K: knowledge, PSL: preliminary security level, LOC: location, TRA: traceability, EXT: extent of damage, SL: security level.

Threat	FR	R	K	PSL	LOC	TRA	EXT	SL
T.SI.Attacker.Malware	SI	3	4	4	0	0	0	4
T.RA.Attacker.DoS	RA	2	2	2	0	0	1	1
T.DC.Attacker.TrafficAnalysis	DC	2	2	2	1	0	1	1

In order to describe the attacker's motivation DIN VDE V 0831-104 introduces railway-specific mitigation factors related to the risk for the attacker to be discovered. The higher this risk for a particular attack, the lower is the motivation to carry it out. This way, existing security controls, can be taken into account.

The standard considers the following mitigation factors: location (LOC), traceability (TRA), and extent of attack (EXT) [2]. LOC determines whether an attack can be executed remotely (LOC = 0) or only given a physical access to the system (LOC = 1). Remote attacks are considered to be more dangerous, as the chance for the attacker to remain undiscovered is higher. Similarly, if an attack can be traced to the attacker (TRA = 1), it is less dangerous than an untraceable attack (TRA = 0). EXT denotes the potential damage of an attack, which can be either *critical* (EXT = 1) or *serious* in case fatalities might be involved (EXT = 0).

This attacker model is applied during the risk assessment to evaluate each potential threat and determine the security level required to protect the system against it (see Section 3.2). Though this approach does not explicitly employ traditional attacker categories including, e.g., basic user, cybercriminals, terrorists, insiders, or nation state [5], it uses the same idea to describe attacker's capabilities.

In the specific context of this paper involving safety aspects, the only relevant goal of the attacker is to cause collisions or derailment of trains. This goal can also be achieved by simply blocking the tracks, exchanging the colors of a light signal or detaching the light signal from the OC and applying current, so that the signal shows clear when it should show stop. For example, political activists from Greenpeace are reported to cause disruptions by sabotaging the railway to prevent the transportation of nuclear waste (http://www.greenpeace.org/international/en/news/Blogs/nuclear-reaction/greenpeace-block-nuclear-waste-transport/blog/35204/). This kind of simple physical attacks targeted against individual field elements have always been possible and cannot be fully prevented by IT security mechanisms.

The digitalization in turn introduces much broader adversarial opportunities due to connectivity and usage of regular IT components. Thus, our goal is to protect the railway signalling network against *one-affects-all* attacks that can be applied to a multitude of field elements at the same time and can paralyze the complete infrastructure. Additionally, by the use of COTS products, attacks can become remotely executable in case the attacker finds a way to access the network shown in Figures 1 and 2. In contrast to such scenarios, physical attacks are much more restricted. As the hardware belonging to Z.ILS and Z.MDM is set up in a building with physical access control, it is reasonable to assume that an attacker cannot gain physical access to this hardware to modify or even replace the components. The OCs in Z.OC are installed in a junction box and therefore physically accessible for the attacker to tamper with the hardware or replace it entirely. However, we assume that physically tampering with an OC is equivalent to local attacks we described before (blocking the tracks; changing colors of light signal). They do not scale to a one-affects-all attack, because each field element has to be attacked individually. Also it is considered virtually impossible to protect against such physical attacks in a large-scale infrastructure like nationwide railway signalling [6, 7]. For this reason, only protection against high-impact one-affects-all attacks is the main scope of this paper.

3.2. Threat Analysis and Preliminary Security Level. We perform a threat-based risk analysis as defined in DIN VDE V 0831-104 by systematically listing cyberattacks that threaten the components presented in our system model. This approach allows us to consider typical cyberattacks as well as infrastructure-specific and railway-specific attacks. As a result of the threat analysis, 67 threats are defined.

Some examples are given in Table 1. The table also shows the underlying attacker model of DIN VDE V 0831-104 as explained in the previous section.

Each threat was given an identifying name and a detailed description, where the description provides details about attack implementation and potential impact. For example, T.SI.Attacker.Malware describes an attacker who introduces malware to undermine the integrity of the railway CCS. A typical denial-of-service (DoS) attack is covered by the threat named T.RA.Attacker.DoS. A threat more typical for railway CCS describes an attacker that records and analyses the traffic of a signalling network in order to prepare a more sophisticated attack (T.DC.Attacker.TrafficAnalysis).

The threats are assigned to at least one of the foundational requirements (FRs) of IEC 62443 being identification and authentication control (IAC), use control (UC), system integrity (SI), data confidentiality (DC), restricted data flow (RDF), timely response to events (TRE), and resource availability (RA).

In addition, each threat was assigned to the zones to which it is applicable. Out of all the 67 threats, 51 are applicable to Z.OC, 51 are applicable to Z.ILS and 47 are applicable to Z.MDM. While 38 threats are relevant for all the three zones (i.e., executable on different components), seven are applicable to exactly two zones, and 22 are specific to a single zone (i.e., exploiting the characteristics of the respective zone).

Table 2: SL vectors of the zones as the result of the analysis.

Zone	IAC	UC	SI	DC	RDF	TRE	RA
Z.OC	3	2	*4*	1	1	3	1
Z.ILS	3	2	*4*	1	1	3	1
Z.MDM	3	2	*4*	1	1	3	1

To estimate the risk imposed by a given threat, we consider the attacker capabilities for this threat in accordance with the attacker model introduced in Section 3.1. The attacker's resource and knowledge capabilities are combined to form a preliminary security level (PSL) related to the given threat. The PSL is later used to calculate the final security level (SL) for the respective threat. A SL—ranging from 1 (low) to 4 (high)—describes the level of protection a system provides against an attacker.

During the execution of the risk assessment process each threat is assigned values for the attacker capabilities (R, K) and values for the mitigation factors (LOC, TRA, EXT) to calculate and respectively adjust the PSL relevant for the respective threat. Examples are shown in Table 1.

3.3. Calculation of Security Levels. As defined by DIN VDE V 0831-104 [2], the final SL, using the PSL and the mitigation factors, is calculated by

$$\mathrm{SL} = \mathrm{PSL} - \max\{\mathrm{LOC}, \mathrm{TRA}, \mathrm{EXT}\}. \quad (1)$$

According to the equation, the value of the PSL is reduced by one, if any of the mitigation factors equals one (corresponding to a logical "or"). This means that a threat that can only be executed locally (LOC) is traceable (TRA) or has only a critical (*critical* in this case, is the less dangerous/severe outcome) extent (EXT) which will reduce the PSL by one level to form the SL.

This calculation is done for each of the 67 identified threats in the seven FRs. To calculate the SL value for the three zones, each is assigned a vector of seven values corresponding to the respective FRs, as shown in Table 2. The seven entries are determined by the maximum SL value over the threats assigned to the respective zone and FR. For each zone, the FR with the greatest SL determines the security level of the zone (set italic in Table 2). For simplicity, we write SL x, when we refer to an SL vector with maximum entry x. In this way, the SL vector of Z.OC $(3, 2, 4, 1, 1, 3, 1)$ yields SL4 for the zone. Respectively do the vectors of Z.ILS and Z.MDM yield an SL of 4.

We identify three decisive threats that determine the security levels. We use them to exemplify how the attacker capabilities and mitigation factors lead to the SL. Two of the decisive threats are responsible for the value of 4 in all three zones. The first decisive threat is the remote execution of malware on the systems in our reference architecture (T.SI.Attacker.Malware, see Table 1). We assigned an attacker with moderate resources (R = 3) and extended knowledge (K = 4) to it, resulting in PSL = 4 for the threat. The assessment of the mitigation factors resulted in the following values: the threat description implies that it is remotely executable (LOC = 0). A skilled attacker is assumed to be able to hide the traces, such that we consider the threat as not traceable (TRA = 0). Also, an attacker with deep knowledge (K = 4) is capable of performing a carefully targeted attack with potentially serious extent (EXT = 0). Thus, none of the mitigation factors apply to reduce the final SL. Subsequently, plugging the mitigation factors and the PSL into (1) yields SL4.

The second decisive threat describes the manipulation of patches such that legitimate processes execute malicious code chosen by the attacker when rolled out to devices through update mechanisms. Analogous considerations for attacker capabilities and mitigation factors as in the previously discussed case apply to this threat leading to SL 4.

Only for zone Z.MDM there is a third decisive threat that results in SL 4. It covers the manipulation of data on the MDM where the firmware of the ILS and OCs is stored and distributed from. This threat poses a high risk, because the firmware can be manipulated remotely at a central point from which it is distributed to unsuspecting network entities. Without further checks, the ILS and OCs of an entire station's signalling network can be compromised. Again, we consider an attacker with R3 and K4 to perform this attack and could not identify an applicable mitigation factor (LOC = 0, TRA = 0, and EXT = 0). Hence, the analysis of this threat also leads to SL 4 for zone Z.MDM.

4. Security Requirements Elicitation

After having derived the SL of each zone, the specific system requirements can be retrieved from IEC 62443-3-3. The standard contains a list of 100 system requirements that are applicable to each zone depending on the identified SL. We evaluated the SL for each FR to select the security requirements from the IEC 62443-3-3 standard. As a result of our risk analysis, we found 69 system requirements that are relevant for our system model.

The requirements of IEC 62443 are on a generic level, as it is a standard for industrial automation and control system (IACS). In order to reduce the complexity of handling 69 relevant requirements and to conserve the railway-specific knowledge gathered while specifying the threats, we choose to explore an additional approach to derive security requirements. For this approach, we elicit requirements with a methodology suggested by Myagmar et al. [8]. The authors propose to take the outcome of the conducted threat analysis as a basis, and transform each identified threat to a requirement. Deriving requirements besides IEC 62443 opens the possibility of responding to railway-specific factors that cannot be reflected in a IACS standard like IEC 62443.

Examples for such railway-specific requirements are R1, R11, and R13 that are shown and discussed later in this section.

In order to derive requirements, we investigate corresponding threats and formulate a statement indicating which behaviour the system must or must not exhibit in order to prevent the respective threat [8]. We mark the threat to be covered by the respective requirement. Alternatively, we declare that the required behaviour was already contained in a previously found statement, so we mark the threat to be covered by the respective requirement. We repeat the addition of requirements until every threat has at least one corresponding requirement. Subsequently, we derive the following security requirements that are explained in more detail afterwards:

R1 The system shall detect unauthorized physical access to its subsystems and/or prevent relevant exploitations of physical access.

R2 The system shall not allow the compromise of a communication key.

R3 The system shall not disclose classified or confidential data (such as access credentials) to any illegitimate user.

R4 The system shall exclude compromised endpoints from communication.

R5 The system shall not use insecure transfer methods.

R6 The system shall not allow any unauthorized user to access an endpoint (e.g., MDM, ILS, and OC).

R7 The system shall not allow unauthorized and unauthenticated communication between endpoints.

R8 The system shall not violate the runtime behaviour requirements.

R9 The system shall allow for the updating of security mechanisms, credentials, and configurations in order to patch known vulnerabilities.

R10 The system shall not allow the execution of unauthorized software instances.

R11 The system shall maintain the transmission system requirements defined in EN 50159.

R12 The system shall provide mechanisms to detect an undesirable system state change and anomalies.

R13 The system shall impede that an unauthorized user can force it into one of the fall-back levels defined by the railway safety process.

R14 The system shall maintain the integrity of software, firmware, configuration, and hardware.

In the following, we discuss those requirements that are specific to railways as they could violate the safety constraints posed by the domain standards.

The physical access detection required by R1 is especially relevant for the railway domain, because OCs are spatially distributed over large areas next to railway tracks and cannot be as well protected as, for example, within factory premises. Therefore, unauthorized physical access to the junction box of the OC has at least to be detected such that further actions can be triggered by, e.g., a security operations center to avoid or mitigate consequences.

In order to keep a railway station operational, R4 requires that compromised endpoints (OCs) can be excluded from the network, such that benign OCs are not affected. An endpoint is considered compromised, if an attacker can remotely control the safety functionality, because the OC accepts the attacker's commands or the attacker controls the safety-critical software on the endpoint. The disclosure of cryptographic keys belonging to an endpoint renders it compromised as well. Physical access to the OC's hardware constitutes a compromise that can be detected if the junction box is opened or the OC is removed from the rack. Physical access to the steered field element is not a compromise as this attack is already possible in current railway infrastructures without digital components. However, physical attacks of this kind do not scale to multiple OCs, if no shared secrets can be gained by the attacker.

Requirement R5 excludes transfer methods such as network protocols that involve cryptographic functions which usage is discourage by institutions like NIST or national agencies for information security. The requirement enforces the usage of communication protocols that do not employ cryptography that is considered broken.

Due to safety reasons, railway signalling networks require a failure disclosure time, which is addressed by R8. Any security measure that influences the network traffic (e.g., message encryption) must not exhaust the network resources such that the network latency exceeds a threshold of 50 ms as specified by railway safety standards and railway operator specifications.

A common requirement from both security and safety perspectives is the robustness of a transmission system against repetition, deletion, insertion, resequencing, corruption, delay, and masquerade of messages. This is specified by EN 50159, a European standard for safety-related communication in transmission systems required to receive admission to operate a railway system. R11 ensures that these requirements are fulfilled and also considered in the design of a security architecture to avoid fulfilling a requirement twice. Some security functionalities might already be available in the system due to fulfilling EN 50159 or can be established by adding only minimal features.

During the design of the security architecture, it must be considered that railway signalling has processes that take effect in case of technical failure in order to maintain operation of the railway system. These fall-back processes involve human interaction and do not provide the full extent of safety and capacity compared to a fully functional, automated interlocking in terms of failure rate. This risk is covered by R13, which requires the security architecture to be designed in such a way that it does not allow an attacker to force the interlocking system into a fall-back state. The attack surface should not be increased by the security architecture compared to attacks with the same effect already possible today, e.g., physically destroying a cable. A security architecture that can force the safety system into a fall-back state enables the attacker to remotely implement a

large-scale DoS attack causing major disruption in the railway transportation system.

The derived requirements will be used to define a security architecture for our system model. The security concept will be outlined in Section 6.

5. Discussion of the Risk Analysis Process

While conducting the risk analysis process according to DIN VDE V 0831-104, we found some shortcomings of the standards that should be addressed. We also make suggestions how the process could be improved. Moreover, we analyse how the suggestions would influence the resulting requirements.

5.1. Disjunction of Mitigation Factors. Equation (1) allows reducing the PSL to obtain the SL if any of the mitigation factors is applicable. This means that a lower security level is found if one of the following conditions for an attack (as the execution of a threat) is met: The attack either can only be performed locally, is traceable to the attacker, or is not expected to cause fatalities. This contradicts intuitive human risk acceptance. A threat that most certainly causes fatalities but is only locally executable leads to at most SL 3 for the associated zone. This is because the PSL will be reduced by 1 through the mitigation factor LOC = 1. An attack with the same type of attacker (same level of knowledge and resources), that is also only locally executable (LOC = 1) and additionally traceable (TRA = 1) but will not cause fatalities (EXT = 1), will lead to the same security level as the previously constructed threat and will require the same security protection as a result. Whether human lives could be lost due to this attack is not valued in this scenario, because the application of one mitigation factor is sufficient to reduce the SL.

To eliminate this contradiction, we suggest reducing the SL only if all three railway-specific mitigation factors apply. This is reflected by (2), where, instead of the maximum, the minimum over the mitigation factors is subtracted from the PSL:

$$\mathrm{SL} = \mathrm{PSL} - \min\{\mathrm{LOC}, \mathrm{TRA}, \mathrm{EXT}\}. \quad (2)$$

We recalculate our threat analysis described in Section 3.3 according to (2) and find now 19 instead of 2 threats with SL 4. Overall, the SL of 54 of the 67 threats is increased by one, leaving 13 unchanged. Since SL 4 threats existed in our previous analysis, the final SL does not change, but has now greater support with 19 threats instead of 3.

5.2. Reduction of Security Level. The standard DIN VDE V 0831-104 does not state whether it is allowed to apply the mitigation factors to a PSL of 2 or 1. Such a reduction would result in zones which have SL 1 and SL 0, respectively. Following DIN VDE V 0831-104, a zone with SL 1 only requires basic protection against unintentional or incidental attacks, which are already covered by safety measures. SL 0 does not require any security protection at all, although an attack might have been found in the previous analysis.

We suggest that the DIN VDE V 0831-104 should exclude mitigating PSL 2 to SL 1 by definition. In our analysis, this would affect 11 threats obtaining SL 2 instead of SL 1. Still, the SL of the entire zone is dependent on the strongest attack (i.e., the threat with highest SL). Therefore, it is unlikely that a threat with SL 1 is decisive for the system.

5.3. Transforming Threats. During the risk analysis of DIN VDE V 0831-104, very specific threats are collected that are based on detailed knowledge of the reference architecture. In the subsequent process of the prestandard, this railway domain-specific information is lost, as it cannot be reflected by the final requirements taken from IEC 62443, which are generic for IACS. We overcome this by deriving requirements directly from the domain-specific threats as described in Section 4. By employing this methodology, we are able to reflect the specifics of railway operation in our security requirements and maintain the knowledge gathered about potential attacks during the threat analysis. Examples of railway-specific requirements are given and explained in Section 4 based on requirements R1, R8, R11, and R13.

5.4. Assigning Maximum Entry in SL Vector. Contrary to our previously described approach, DIN VDE V 0831-104 suggests assigning the maximum entry of the SL vector to all FRs it defines as being safety relevant. The safety-relevant FRs are IAC, UC, SI, and TRE [2]. With this method our SL vector for `Z.OC` would be adjusted from $(3, 2, 4, 1, 1, 3, 1)$ to $(4, 4, 4, 1, 1, 4, 1)$ (compare Table 2). This transformation aims to reduce the complexity in SLs, because only one entry appears (besides the values defaulting to "1"). However, we believe that this adjustment of the level is not justified by the achieved simplification, as it increases the number of requirements derived from it. Applied to our analysis, the transformation of the SL vector results in 15 additional requirements compared to the 69 we identified with the SL vectors explained in Section 3.3. For instance, UC = 2 is increased to UC = 4, which results in 12 additional requirements, stemming from this FR. The additional requirements entail the implementation of further measures and additional time and financial effort that could otherwise be avoided.

In hindsight, the comprehensibility of the process' result is lost, because it is unclear whether an entry in the vector stems from an actual threat demanding it or from adjusting the values to the greatest entry. Therefore, we chose not to follow the suggestion and kept the vectors as presented in Section 3.3.

6. Software and Hardware Architectural Considerations for Safety and Security

We use the security requirements derived in Section 4 to define a hardware-based security platform for railway CCS.

In this article, we analyse the coexistence of safety and security applications that not only have different assurance levels but should be certified by different authorities competent in each domain. In our case from safety and information security authorities, most countries operate a

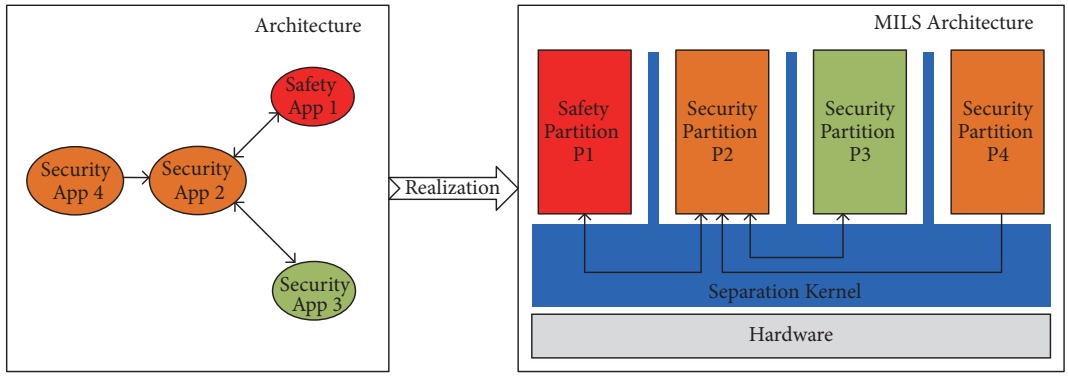

Figure 3: Proposed architecture and its realization using a Separation Kernel.

national safety authority whose admission is required to be allowed to operate a railway system (safety case). Most of the safety authorities and railway operators recognize the increasing need for security functionality and respective security certification due to the utilization of COTS products in the domain. As most safety authorities lack the experience for security certification, this task is fulfilled by respective national authorities for information security.

6.1. The Object Controller as a Mixed-Criticality System. With regard to the coexistence of safety-critical and security applications on the same platform, the railway safety standard EN 50128 requires that all software should be certified to the highest safety integrity level (SIL) unless evidence of independence (freedom of interference) can be provided [9, Section 9.4.9]. However, for multiple reasons, we do not want to treat the newly developed security applications as belonging to SIL4 like the OC's safety application. Obtaining the highest safety certification is hardly feasible for security applications since they are developed with a dynamic lifecycle, considering the ever changing attack potential of the environment that evolves much faster than any safety certification cycle. Certifying the safety-critical subsystem without the need to evaluate the security applications allows us to protect the safety system with complex security measures that can react to the changing vulnerability landscape in an adequate time frame without recertification.

Thus, from the safety certification point of view, we are building a mixed-criticality system that combines applications with different safety assurance levels on the same (hardware) platform [10]. In our case, the OC safety application with SIL4 requirements and security applications with no safety assurance requirements are combined on the same platform.

6.1.1. MILS Architecture as a Solution for the Sharing Challenge. We follow the Multiple Independent Levels of Security (MILS) approach [11–14] in order to simplify the certification described in the previous section. MILS is a high-assurance safety/security architecture based on the concepts of separation and controlled information flow. The cornerstone of the architecture is a separation mechanism that encapsulates applications of different criticality in different partitions. These partitions reduce mutual dependencies to communications over channels explicitly defined by policies. The separation mechanism has to be nonbypassable (cannot be circumvented), evaluable (small and simple enough that proof of correctness is practical and affordable), always invoked, and tamper-proof (cannot be modified without authorization) [14]. The safety and security assurances for this separation mechanism are described in Section 6.1.2

The MILS platform is implemented as a software-hardware system, consisting of a software separation kernel (SK), the hardware central processing unit (CPU), and memory management unit (MMU), as well as other critical hardware devices (e.g., I/O MMU) and their software (see Figure 3).

The central part of a MILS platform is its SK, which provides separated partitions, manages hardware, and enforces security policies for information flow, access control, and resource availability. A SK is a special kind of operating system often based on the microkernel approach [15] to achieve modularized design and minimize the trusted computing base. These properties allow for a formal verification of the kernel [16]. MILS is based on system decomposition of hardware and software into meaningful and—from a security point of view—atomic components. Thus, safety and security by design is achieved by partitioning the system on the architecture level, using the SK to enforce these architectural decisions.

The MILS terminology defines *separation* as the condition where two applications are allocated in different partitions and there is an explicit communication channel between these partitions. Partitions are used to create security domains. Communication between partitions represents the explicit crossing of domain boundaries and is realized using communication objects under the control of the SK. The information flow policy used between partitions using the communication objects is statically defined during the integration phase by specifying the allocation of communication objects to the partitions and the permissions. During runtime, the SK will enforce the allocation and access permissions as per the static configuration.

For realizing our use-case of integrating the safety and security applications on the same platform, we map

these applications to different partitions. This architectural refinement is shown in Figure 3. Depicted on the left-hand side is a high level architecture, showing the high-critical safety application in red and security applications from different domains in green and orange. The arrows between the applications indicate the communication channels and their directions. The architectural refinement using the SK is shown on the right-hand side, where the safety and security applications are assigned to safety and security partitions and the communication channels are realized using the communication objects that are under control of the SK.

With the help of the underlying SK, we make sure that the safety partition is both spatially and temporally separated from the security partitions, in order to guarantee code and data integrity as well as real-time. The hardware platform used for deploying the proposed architecture shall be fast enough to reserve CPU time required for the safety application to meet its deadlines and at the same time have remaining CPU time available to the security applications to perform their functions.

With such a well-separated MILS architecture, we keep the existing safety certification of the safety application once it is integrated together with the security applications in our new system. We pave the way to retain the safety certification when the security applications are updated in-the-field to respond to the changes in the security landscape, such as the discovery of a weakness in a cryptographic algorithm or a vulnerability in the deployed software.

6.1.2. Safety and Security Assurance for the Separation Kernel Subcomponent. As described in Section 6.1.1, the main function of SK subcomponent in the MILS architecture is to provide strong separation and well-defined information flow channels between partitions that host applications of different criticality levels. The SK can be considered as a single point of failure in the MILS architecture. Thus, the SK shall provide assurance for the highest safety and security levels required for the system.

This is described in the EN 50128 standard that states that the subcomponent that provides separation between components shall be certified at the same level of the highest assurance application. Thus, the SK we use shall be certifiable to SIL4 which corresponds to the assurance level required by the safety application.

For the security assurance of SK, we rely on Common Criteria (CC) [17] which is a well-established international standard for security evaluation and certification. The CC approach demands the developer of a product to conduct an asset identification, threat analysis describing countermeasures in the form of security objectives and deriving of the security functional requirements (SFRs) that fulfill the security objectives followed by the implementation, validation, and verification of the SFRs.

The IT life-cycle described in railway security prestandard DIN VDE V 0831-104 is comparable to these steps in CC. For instance, the procedure description matches how a CC security target for a product is developed.

For the CC threat analysis, the assets correspond to the resources (such as memory, CPU cores, files, interrupts, and CPU processing time) assigned to a partition and the applications belonging to different partitions are considered as attackers. The SFRs ensure that the confidentiality, integrity, and resource availability properties of the assets protected by the SK are preserved from an application belonging to a different partition according to the partitioning and information flow security policies. Thus, the SFR claims of the SK are highly relevant for its role in the MILS architecture, namely, providing mechanisms for separation and well-defined information flow channels between partitions. The detailed description of threat analysis, SFRs, and its validation and verification is described in the SK CC assurance artefacts.

DIN VDE V 0831-104 assumes that IT security products purchased on the market already have IEC 62443 compatible certification [2] and the ISASecure's interpretation of IEC 62443-4-1 suggests that COTS operating systems should at least be certified to CC EAL3 [18]. There are also ongoing discussions on how to combine CC and IEC 62443 (e.g., in the certMILS project [19]). The current understanding is that CC is used on subcomponents such as operating systems and IEC 62443-4-1 and IEC 62443-4-2 on integrated devices (components) designed to be used as part of the infrastructure (system) compliant to IEC 62443-3-2 and IEC 62443-3-3.

It is also important to highlight that assurance level (EAL) and SFRs are completely independent; i.e., there can be very weak security functionality and high assurance as well as very strong security functionality and assurance level that just fulfills the purpose. Moreover, from business point of view, the cost of evaluations grows exponentially [20] with higher EALs, and thus, any system designer would need to look for approaches to manage these costs by choosing the right assurance level. The French national security authority (ANSSI) also defines three levels of assurance with EAL3+ being a medium one [21].

Based on this state-of-the-art research, we have chosen the EAL3+ as the very minimum for the security assurance for the SK subcomponent while requiring very strong functional security requirements (i.e., separation) in the security targets of SK.

6.2. Security Applications for Integrity Protection and Resilience. In order to define our security architecture that fulfills the requirements defined in Section 4, we utilize the shell concept of Schlehuber et al. [22]. In the shell concept, safety is protected by a shell of security measures. Any communication with the safety part passes through the security shell. Thus, an attacker must successfully penetrate the security shell in order to attack the safety functionality.

The OCs are currently the weakest spot of the CCS architecture, as they are not protected by a surrounding building. Thus, we propose a security architecture for OCs in our system definition (see Figure 2). We call an OC that implements our proposed security architecture, as illustrated in Figure 4, a Secure Object Controller (SecOC). The depicted building blocks are discussed in this section. How

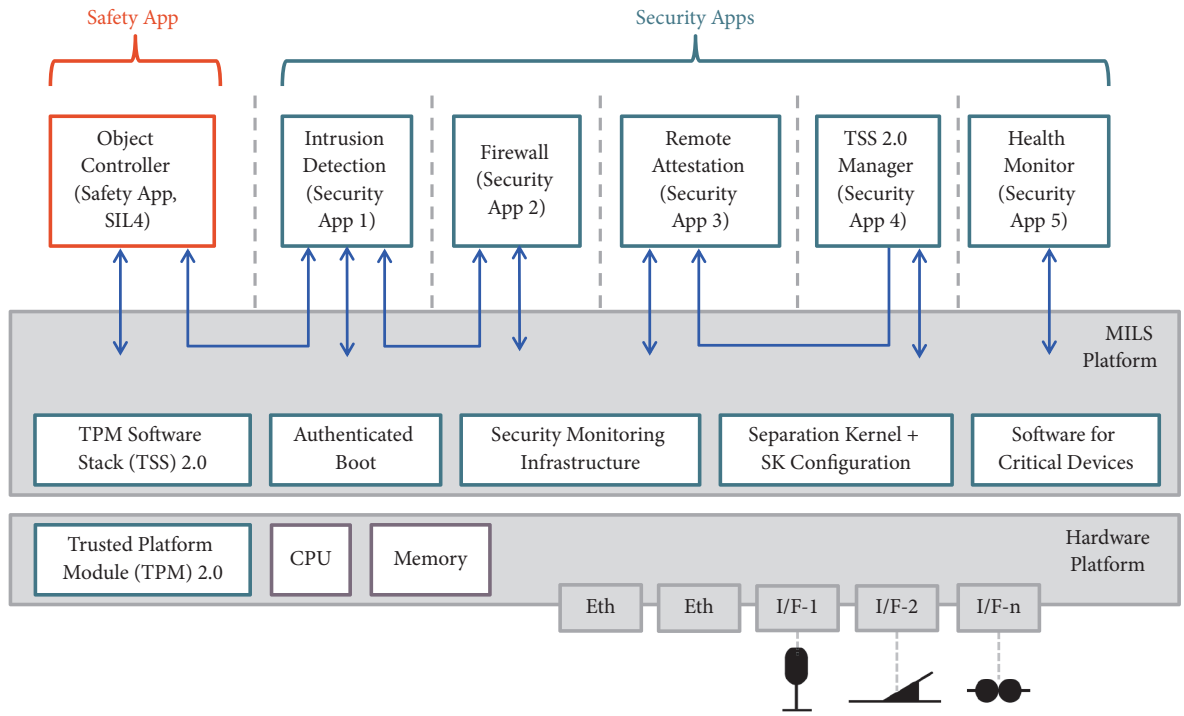

Figure 4: Hardware-based security architecture for the Secure Object Controller.

each building block contributes to the fulfillment of the security requirements is discussed in Section 7.

Hardware Platform. The SecOC is located remotely at railway tracks. To ensure that this safety-critical system is not manipulated by malicious attackers, a continuous proofing is performed by using trusted computing technologies [23]. As a hardware root of trust (ROT), a Trusted Platform Module (TPM) 2.0 specified by the Trusted Computing Group (TCG) [24] is used. It provides secure storage, secure execution for cryptographic operations, and additional features, e.g., facilitating authenticated boot and secure update procedures. The TPM is a special type of a hardware security module. In our case, it is implemented as a dedicated chip. Petri et al. showed that cryptographic agility, enhanced authorization, and modularity of the TPM 2.0 specification make it suitable to secure long-lived resource-constrained systems like controllers [25]. Authenticated boot and remote attestation build on the capability of the TPM to persistently store the status of the loaded software as so-called configuration measurements (hash-chains) within the TPM's Platform Configuration Registers (PCRs) [26].

To further protect the SecOC against physical attacks, the housing or chassis of the hardware platform raises an alert every time it is opened. The alerts are written into the TPM via the BIOS and can only be reset by authorized personnel. Chassis open alerts work even if the SecOC is disconnected from power supply and make physical interference from an attacker evident.

For the redundant network connection, the hardware platform provides two Ethernet interfaces. In order to steer the field elements, wired interfaces to signals, point machines, and train detection systems are designated.

Software (MILS) Platform. The software platform is constructed following the MILS methodology [14]. The security core of the proposed architecture builds on the SK. The MILS platform architectural layer provides a controlled interface to the hardware and enforces security policies for information flow, access control, and resource availability.

The TPM Software Stack (TSS) 2.0 specified by the TCG [24, 27, 28] provides an API for TPM 2.0 to applications and is also part of the platform. A software for critical devices controls direct access of hardware to the main memory without involvement of CPU and MMU that would otherwise bypass the SK. A Security Monitoring Infrastructure module inside the MILS platform collects information about system state and network traffic required for detecting the network anomalies and performing health monitoring.

Security Applications. Security functions necessary for the SecOC to fulfill the identified security requirements are realized by security applications (denoted as *Security App* in Figure 4) running on the MILS platform. This includes applications enabling the protection of the SecOC software integrity at boot- and runtime, integrity reporting, remote attestation, and secure software update as well as health monitoring, network traffic filtering, and intrusion detection (see Section 6.2).

Safety Applications. Alongside the security applications, one or more safety applications run on the MILS platform. In case

of the SecOC, we encapsulate the existing functionality of an OC (see Figure 1) in the safety application that operates a point or a signal.

6.2.1. Hardware Trust Anchor and Security Protocols. A TPM 2.0 provides a hardware root of trust (ROT). Using an authenticated boot procedure, the platform's ROT for Reporting generates a report about the boot process by creating evidence about the integrity of the booted software components. Using a remote attestation procedure, this evidence can be conveyed to an external verifier in order to appraise the evidence.

In our architecture, we will use the Time-Based Uni-Directional Attestation (TUDA) protocol [29] as remote attestation procedure. In contrast to traditional remote attestation procedures (e.g., [30]), the attestor (or the verifier, respectively, depending on which entity initiates the procedure) does not need to send a nonce, in order to prove the freshness of the created evidence, but uses trusted timestamps originating from an external trusted source of time. This approach has the advantage of decoupling the activities of creating the integrity evidence, conveying it and then appraising it via a verifier, therefore enabling the appraisal of past states and also reducing the utilization of the TPM significantly.

6.2.2. Authenticated Boot Security Architecture. In order to create evidence that enables integrity proving of a SecOC and its corresponding software components that compose the runtime environments, an authenticated boot procedure is introduced. Even before booting, integrity measurements—resilient hash-values of every software component involved, including the BIOS—are created and aggregated in a corresponding partition of the SecOC. Measurements of safety applications and their configuration data are created with the help of the introduced authenticated boot kernel component (see Figure 4) on startup and during observation of update processes. The separation enforced by the SK guarantees that the creation of integrity evidence, as a basis for remote attestation procedures, has no impact on the runtime behaviour of the safety applications.

The relationships of interfaces among the entities in the SecOC are illustrated in Figure 4. Basically, every higher level interface with respect to creation of evidence is based on the TSS 2.0. Secure communication channels between partitions are realized using communication objects that are governed by the SK. The SK provides the separation assurance required to shield the safety-related partitions from malicious or unintentional harmful use of these communication objects.

6.2.3. Runtime Integrity Monitoring. Considering that rebooting or updating an OC does not happen frequently, runtime mechanisms for integrity control and system resilience are required to fulfill the security requirements. This functionality is provided by a dedicated health monitor (see Figure 4) that continuously performs collection of data regarding application state, analyses this data for changes or anomalies and does reporting [31, 32]. For this purpose, the communication objects of the SK need to be extended with monitoring capabilities to detect failures or attacks in system safety or security services. Another data source for the health monitor is the kernel-level security monitor. The health monitor can be noninvasive or enable invoking specified resilience mechanisms similar to the ones proposed in [33, 34] desired for some particular scenarios. Thus, to reduce the service downtimes, a SecOC may be able to recover after failures or attacks, e.g., by automatically resuming to a known valid configuration instead of failing. In this case, this behaviour does not have any impact on the assurance level of the safety applications because the application itself is not changed by the health monitor.

6.2.4. Secure Update. Software, firmware, and configuration updates are necessary to fix software bugs and vulnerabilities in the SecOC. The new features introduced by the TPM 2.0 specification, such as monotonic counters and enhanced authorization, make it possible to protect system data, e.g., safety-relevant settings and to ensure that only updates from authorized sources are accepted and downgrade attacks are prevented [26].

6.2.5. Intrusion Detection. The Intrusion Detection System (IDS) provides a defense mechanism against network-based attacks and applies after the network traffic has been initially filtered by a firewall. Types of network traffic are distinct and invariable. Protocols and ports for control commands, secure update, and remote attestation are known in advance. The firewall can be configured by a predetermined whitelist. By collaborating among multiple SecOC instances in a defined area of the controlled field elements, the IDS is enabled to detect adverse commands, dangerous infrastructure configurations, and misuse on application level. Our IDS solution allows the SecOCs to validate received commands as an additional layer of defense beyond authenticated communication channels. Ability current OCs do not possess. The IDS is fine-tuned to the CI's network topology (i.e., the track layout and the position of signals and points) and the utilized transport and application protocols. In this way, the IDS is enabled to leverage context information of the controlled infrastructure to enhance the intrusion detection accuracy. In a second step, counteractions on detected intrusions are defined that respect the safety functions of the CI. We carefully design the intrusive counteractions such that they do not alter the network channel properties beyond the specification of latency and message loss that the safety application is anyway required to tolerate.

7. Coverage of Security Requirements

In the following, we argue that all security requirements defined in Section 4 are covered by the proposed hardware-based security architecture for the SecOC. Once a working prototype of our architecture is built, we evaluate it against the requirements with the proposed coverage.

R1 *The system shall detect unauthorized physical access to its subsystems and/or prevent relevant exploitations of physical access.*

Attack scenarios of physical access are addressed in several ways. General access to the chassis of the SecOC can be detected through the chassis open alert provided by the hardware platform (see Section 6.2). The access is reported using the authenticated TPM attestation means. Also the alteration of the main firmware of the device or the replacement of those hardware components that carry an Option-ROM can be detected by the attestation. Using the audit capabilities of the attestation protocol, even physical access to the device or management backend cannot alter postincident logs in order to, e.g., fake the firmware or configuration status of the device during an accident.

Additionally, the devices prevent the cloning of a devices identity, since the relevant data is stored inside the TPM. Though a TPM and the associated identity can be stolen, they cannot be duplicated. The use of monotonic counters inside the TPM also prevents so-called rollback attacks of local firmware, where older versions are used in place of more recent versions.

Please note that some physical manipulations are still possible, such as the tapping in between CPU and TPM. Even though concepts against such attacks exist (e.g., RayzorClam) they are not needed, since the same attacker class can reach results of direct I/O manipulation or hardware replacement much cheaper, than in-field PCB modifications.

R2 *The system shall not allow the compromising of a communication key.*

The TPM 2.0 provides a secure storage for communication keys and any other cryptographic keys as well as a secure execution environment for cryptographic operations to protect them from compromise.

R3 *The system shall not disclose classified or confidential data (such as access credentials) to any illegitimate user.*

Service access, user login, and privilege escalation events can be stored in the TPM in order to create evidence about system state that may enable unauthorized disclosure and can result in actions ranging from event notification to automated remediation or quarantine procedures. The TPM 2.0 provides a secure storage for cryptographic keys, a secure execution environment for cryptographic operations, and supports state-of-the-art cryptographic algorithms to protect confidential data from unauthorized disclosure.

R4 *The system shall exclude compromised endpoints from communication.*

Continuous proving of platform integrity and attestation allows detecting and excluding compromised endpoints from the communication. Using a suitable integrity measurement architecture policy, those attestations can validate the integrity of code as well as data elements of the platform. The proposed architecture will measure all complete static partitions, including code and data segments. Additionally, an interface exists, that these measured (and well-behaved) partitions will in turn use to measure their dynamic (runtime) configuration data.

The concept of measuring code as well as data and doing so context dependently is not unusual; the default Linux policy will measure all code (executables and libraries loaded by all users). For the root user, it will additionally measure all data files opened. For example, measured runtime data can be the state of the controlled field element (e.g., signal aspect and direction of the point). The states are known in advance so that reference measurements can be created.

R5 *The system shall not use insecure transfer methods.*

Our architecture does not use transfer protocols with known vulnerabilities. Update mechanisms are applied to react to newly discovered vulnerabilities (see R9).

R6 *The system shall not allow any unauthorized user to access an endpoint.*

Access to the endpoints is protected by user credentials.

R7 *The system shall not allow unauthorized and unauthenticated communication between endpoints.*

The TPM 2.0 can be used to equip each OC with a secure and unique identity that provides the basis for secure networking. Allowed network flows are assessed and enforced by a firewall.

R8 *The system shall not violate the runtime behaviour requirements.*

Separation mechanisms supported by the SK allow statically assigning user applications all resources needed for fulfilling its runtime requirements and provide guarantees that the safety application and security applications do not have to compete with each other for resources.

R9 *The system shall allow for the updating of security mechanisms, credentials, and configurations in order to patch known vulnerabilities.*

The TPM 2.0 based secure update mechanism enables remote updates for security mechanisms, credentials, and configurations and rollback protection.

R10 *The system shall not allow the execution of unauthorized software instances.*

Depending on the criticality of the software instance affected, an authenticated boot procedure enables countermeasures ranging from enabling restrictions on execution privileges to pausing affected partitions. A fine-granular action catalogue ensures that countermeasures cannot affect functions of a safety-critical partition.

R11 *The system shall maintain the transmission system requirements defined in EN 50159.*

An implementation of the architecture will be evaluated to assess that the requirements are not violated.

R12 *The system shall provide mechanisms to detect an undesirable system state change and anomalies.*

Based on the collected integrity evidence, changes in the boot system configuration can be revealed. The health monitor and the intrusion detection analyse the runtime behaviour of the applications and detects undesired states.

R13 *The system shall impede that an unauthorized user can force it into one of the fall-back levels defined by the railway safety process.*

The applications of the security part of the shell concept proposed by Schlehuber et al. [22] work free of interference with the applications in the safety part of the shell. Without substantial and well-defined reason, the security does not trigger fail-safe states. The security architecture provides resilience mechanisms that help to recover the system from attacks and reduce downtimes.

R14 *The system shall maintain the integrity of software, firmware, configuration, and hardware.*

Boot and runtime integrity control of software, firmware, and configuration together with resilience mechanisms help to protect the system from compromise. Depending on the hardware platform, the hardware integrity, a kernel interface conducting the respective measurements or platform certificates with hardware configuration attributes, is used to maintain the hardware integrity. Hardware integrity refers to the unmodified assembly of hardware to a composite device (i.e., the SecOC).

As our research is based on DIN VDE V 0831-104 which is a guideline to apply IEC 62443 to the railway domain, we believe that we sufficiently covered all security requirements. We are aware that it is hard to know when a list of security requirements is complete, as security—contrary to safety—is an iterative process. This is why our architecture contains the ability to be extended by further security measures if necessary.

8. Conclusion

In this paper, we report our practical experience obtained while executing a security requirements engineering process for railway signalling systems in a real-world scenario. This process is defined in the German prestandard DIN VDE V 0831-104 which must be conformed to in order to receive admission to operate railway infrastructure. Our reference architecture covers the components required to operate the track-side equipment necessary for safe train journeys, being signals and points. The reference architecture was translated by experts to a system definition featuring zones. Our risk analysis yielded a SL of 4 for the identified zones demanding the highest security requirements according to IEC 62443, a security standard for industrial control systems. Overall, the process is suited for finding useful system requirements, especially because it is constructed along already established safety and security standards. However, we came across some minor shortcomings that we presented in this paper. We proposed modifications to overcome them and analysed the impact of our suggestions on our real-world application.

The derived requirements were used to design a security architecture for safety-critical railway signalling networks based on a hardware platform allowing to run mixed-criticality applications. We propose the security architecture based on the MILS concept to comply with the identified security requirements. The concept is implemented using a separation kernel that controls resource sharing between safety and security applications. The security applications of our architecture employ health monitoring, authenticated boot, remote attestation procedures, secure updates, and anomaly detection services. These capabilities provide a wide spectrum of security-related information including complementary indicators that improve security assessments as well as cryptographic integrity evidence that can be conveyed to management systems to prove that a device is a trustworthy system. Additionally, secure audit logs are created that are able to document past states a device was in with the highest level of assurance. We analyse how the proposed security architecture fulfills the requirements we derived.

We will implement and demonstrate the feasibility of the proposed security architecture during the HASELNUSS project (https://haselnuss-projekt.de/index_en.php). The project aims to show that it is possible for safety and security applications to coexist in the safety environment in order to provide safe and secure railway operation. During the course of the project, we will evaluate our platform regarding the security requirements and work towards safety certification.

Data Availability

There is no underlying data to publish.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work presented in this paper has been partly funded by the German Federal Ministry of Education and Research (BMBF) under the project "HASELNUSS: Hardwarebasierte Sicherheitsplattform für Eisenbahn-Leit- und Sicherungstechnik" (ID 16KIS0597K).

References

[1] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliability Engineering & System Safety*, vol. 139, pp. 156–178, 2015.

[2] "DKE: Elektrische Bahn-Signalanlagen – Teil 104: Leitfaden für die IT- Sicherheit auf Grundlage der IEC 62443 (DIN VDE V 0831-104)," 2015.

[3] S. Islam, N. Suri, A. Balogh, G. Csertán, and A. Pataricza, "An optimization based design for integrated dependable real-time embedded systems," *Design Automation for Embedded Systems*, vol. 13, no. 4, pp. 245–285, 2009.

[4] J. Braband, "Towards an IT security risk assessment framework for rail- way automation," CoRR abs/1704.01175, http://arxiv.org/abs/1704.01175, 2017.

[5] M. Rocchetto and N. O. Tippenhauer, *On attacker models and profiles for cyber-physical systems*, Springer International Publishing, 2016.

[6] APTA Control and Communications Working Group, "Securing Control and Communications Systems in Rail Transit Environments – Part II: Defining a Security Zone Architecture for Rail Transit and Protecting Critical Zones," Tech. rep., APTA, 2013, http://www.apta.com/resources/standards/documents/apta-ss-ccs-rp-002-13.pdf, APTA-SS-CCS-RP-002-13.

[7] H. Bock, J. Braband, B. Milius, and H. Schäbe, "Towards an IT Security Protection Profile for Safety-Related Communication in Railway Automation," in *Computer Safety, Reliability, and Security*, pp. 137–148, Springer, 2012.

[8] S. Myagmar, A. J. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in *Symposium on Requirements Engineering for Information Security (SREIS)*, vol. 2005, pp. 1–8, 2005.

[9] "CENELEC - European Committee for Electrotechnical Standardization: EN50128 - Railway applications — Communications, signalling and pro- cessing systems — Software for railway control and protection systems. No. EN 50128:2001 E, CENELEC Central Secretariat, rue de Stassart, 36, B- 1050 Brussels," 2010.

[10] S. Baruah, H. Li, and L. Stougie, "Towards the design of certifiable mixed- criticality systems," in *Proceedings of the Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE*, pp. 13–22, IEEE, 2010.

[11] J. A. Foss, P. W. Oman, C. Taylor, and W. S. Harrison, "The MILS architecture for high-assurance embedded systems," *International Journal of Embedded Systems*, vol. 2, no. 3/4, p. 239, 2006.

[12] K. Netkachova, K. Müller, M. Paulitsch, and R. Bloomfield, *Security-Informed Safety Case Approach to Analysing Mils Systems*, 2015.

[13] J. M. Rushby, "Design and Verification of Secure Systems," in *Proceedings of the Eighth ACM Symposium on Operating Systems Principles, SOSP '81*, pp. 12–21, ACM, New York, NY, USA, 1981.

[14] S. Tverdyshev, H. Blasum, B. Langenstein et al., "MILS Architecture," *EURO-MILS*, 2013.

[15] G. Heiser and K. Elphinstone, "L4 microkernels: The lessons from 20 years of research and deployment," *ACM Transactions on Computer Systems*, vol. 34, no. 1, pp. 1:1–1:29, 2016.

[16] G. Klein, J. Andronick, K. Elphinstone et al., "Comprehensive formal verification of an OS microkernel," in *ACM Transactions on Computer Systems*, vol. 32, pp. 2:1–2:70, 2014.

[17] "Common Criteria Sponsoring Organizations: Common criteria for information technology security evaluation. version 3.1, revision 5," http://www.commoncriteriaportal.org/cc/, 2017.

[18] ISA Security Compliance Institute, "SDLA-312 Security Development Life- cycle Assessment Version 3.0," http://www.isasecure.org/en-US/Certification/IEC-62443-SDLA-Certification, 2014.

[19] "certMILS project - H2020 project reference: 731456: certmils - compositional security certification for medium- to high-assurance cots-based systems in environments with emerging threats," http://www.certmils.eu/.

[20] G. C. Wilshusen, "Information Assurance: National Partnership Offers Ben- efits, but Faces Considerable Challenges," Tech. rep, United States Government Accountability Office, Washington, DC, USA, 2006, http://www.gao.gov/new.items/d06392.pdf.

[21] "ANSSI: Agence nationale de la sécurité des systèmes d'information: Anssi: Qualification de produit," https://www.ssi.gouv.fr/entreprise/qualifications/qualification-de-produit/.

[22] C. Schlehuber, M. Heinrich, T. Vateva-Gurova, S. Katzenbeisser, and N. Suri, "Challenges and approaches in securing safety-relevant rail- way signalling," in *Proceedings of the Security and Privacy Workshops (EuroS&PW) 2017 IEEE European Symposium on*, pp. 139–145, IEEE, 2017.

[23] N. Asokan, J. Ekberg, K. Kostiainen et al., "Mobile Trusted Computing," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1189–1206, 2014.

[24] Trusted Computing Group, *Trusted Platform Module Library Specification, Family 2.0, Level 00, Revision 01.38 edn*, 2016.

[25] R. Petri, M. Springer, D. Zelle et al., "Evaluation of lightweight TPMs for automotive software updates over the air," in *Proceedings of the 4th Embedded Security in Cars (escar)*, USA, 2016.

[26] A. Fuchs, C. Krauß, and J. Repp, "Advanced Remote Firmware Upgrades Using TPM 2.0," in *Proceedings of the 31st International Conference on Information Se- curity and Privacy Protection IFIP SEC 2016*, pp. 276–289, Springer International Publishing, 2016.

[27] "TSS system level API and TPM command transmission interface specification," http://www.trustedcomputinggroup.org/resources/tss_system_level_api_and_tpm_command_transmission_interface_specification, 2016.

[28] "TSS TAB and resource manager," http://www.trustedcomputinggroup.org/resources/tss_tab_and_resource_manager, 2016.

[29] A. Fuchs, H. Birkholz, I. McDonald, and C. Bormann, "Time-Based Uni- Directional Attestation," in *Internet-Draft draft-birkholz-i2nsf-tuda, The Inter- net Engineering Task Force (IETF)*, 2017, https://datatracker.ietf.org/doc/draft-birkholz-i2nsf-tuda/.

[30] R. V. Steiner and E. Lupu, "Attestation in wireless sensor networks: A survey," *ACM Computing Surveys*, vol. 49, no. 3, pp. 51:1–51:31, 2016.

[31] A. M. Azab, P. Ning, E. C. Sezer, and X. Zhang, "HIMA: A hypervisor- based integrity measurement agent," in *Proceedings of the Computer Security Applications Conference, 2009. ACSAC '09. Annual*, pp. 461–470, 2009.

[32] B. D. Payne, M. Carbone, and W. Lee, "Secure and flexible monitoring of virtual machines," in *Proceedings of the 23rd Annual Computer Security Applications Conference*, ACSAC, 2007.

[33] M. Dinkel, S. Stesny, and U. Baumgarten, "Interactive self-healing for black-box components in distributed embedded environments," in *Proceedings of the Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference*, pp. 1–12, 2007.

[34] D. Garlan, S. Cheng, A. Huang, B. Schmerl, and P. Steenkiste, "Rainbow: architecture-based self-adaptation with reusable infrastructure," *The Computer Journal*, vol. 37, no. 10, pp. 46–54, 2004.