MODEL-BASED SYSTEMS ENGINEERING USING SECURITY DESIGN PATTERNS IN THE CONTEXT OF ISO/SAE 21434

Japs, Sergej (1);
Faheem, Faizan (2);
Anacker, Harald (1);
Husung, Stephan (2);
Dumitrescu, Roman (3)

1: Fraunhofer Research Institute for Mechatronic Systems Design IEM;
2: Technische Universität Ilmenau;
3: Universität Paderborn

ABSTRACT

The development of modern vehicles is complex, especially regarding compliance with security and safety. ISO/SAE 21434 considers security and safety along the entire product life cycle. According to the standard, a system architecture, a risk analysis, and the application of countermeasures are carried out in the early system design.

Design patterns are solutions to known design problems. Security Design Patterns (SDP) describe countermeasures and are used to reduce risk. After our literature review, we did not find a suitable approach that presents SDPs that would be applicable in early system design.

In this paper, we present 10 SDPs for early system design, which we evaluated during an 11-week student project with 28 teams. We present the results of the quantitative analysis and the evaluation of the feedback.

Keywords: ISO/SAE 21434, Systems Engineering (SE), Knowledge management, Risk management, Case study

Contact:
Japs, Sergej
Fraunhofer Research Institute for Mechatronic Systems Design IEM
Germany
sergej.japs@iem.fraunhofer.de

Cite this article: Japs, S., Faheem, F., Anacker, H., Husung, S., Dumitrescu, R. (2023) 'Model-Based Systems Engineering Using Security Design Patterns in the Context of ISO/SAE 21434', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.268

1 INTRODUCTION

The development of modern vehicles leads to increasingly complex systems. A new highlight of this trend are connected, cooperative and autonomous mobility systems (CCAM). One example of a CCAM is Platooning. This allows multiple vehicles to drive behind each other at a very close distance with the help of a technical control system, without compromising road safety. Considering not only the internal complexity of a vehicle, but also the interaction of multiple vehicles and the infrastructure, this is an extremely complex System-of-Systems (SoS). The development of SoS requires the collaboration of experts from different disciplines (Gausemeier et al., 2014). Model-Based Systems Engineering (MBSE) supports in building the common understanding of the system between the domain experts but also people outside the domains and cross-sectional areas starting from the problem space analysis (Husung et al., 2021) which is part of the concept phase. However, ensuring security and safety is a major challenge. In the automotive industry, ISO 26262 is the de facto industry standard with regard to functional safety. However, this standard does not explicitly consider security. UNECE R155 defines requirements for vehicle security against cyber attacks and must be legally fulfilled from 2024 (TÜV, 2022). Here, ISO/SAE 21434 serves as a guideline for the implementation of UNECE R155. ISO/SAE 21434 requires the creation of a system architecture and defines requirements for a comprehensive risk analysis to be carried out already in the concept phase. Based on the identified risks in the system architecture, countermeasures must be identified in the concept phase. UNECE R155 lists 24 countermeasures against cyber attacks. The result of the concept phase is an initial vehicle system architecture. The work in the concept phase is characterized by the cooperation of several experts in workshops (Japs, 2021). In order to implement countermeasures in the initial system architecture, the textual listing of countermeasures from UNECE R155 is not sufficient. More comprehensive information for countermeasures describing the problem to be solved is needed. Based on our project experience, models of countermeasures support the redesign of the initial system architecture. Suitable tools are Security Design Patterns (SDP). SDPs are established solutions to recurring security design problems. To enable an interdisciplinary team to apply such design patterns in workshops, the descriptions must be generally understandable and the models must consist of simple elements of a modeling language. According to our literature review (cf. Section 2), there are no suitable approaches or sources for design patterns for the automotive domain, which can be applied in early system design by an interdisciplinary team of experts. *How must SDPs be defined so that they can be used during development in the interdisciplinary team to define security counter measures?* In this work, we present 10 such SDPs for the automotive domain (cf. Section 3). We report on the use of our design patterns based on an 11 week project with 140 master students (cf. Section 4).

2 ANALYSIS OF RELATED APPROACHES

In this section, we present our analysis of papers that are related to SDPs (cf. Figure 1). We evaluate the approaches based on a literature review on four requirements. *R1: The description of the design patterns is kept general and only requires basic knowledge of specific security terms.* The concept phase is characterized by collaboration among stakeholders from different disciplines. In this phase, stakeholders are represented who often have a leading position. These stakeholders have a broad expertise and act across departments and companies. According to our project experience, such stakeholders do not have enough detailed knowledge in the area of security. *R2: The design patterns use simple constructs of a modeling language.* This is necessary because in the concept phase only some stakeholders are familiar with modelling system architectures with a standardised modelling language. *R3: The design patterns need to include solutions for security threat resolution.* This establishes alignment with UNECE R155 and ISO/SAE 21434. *R4: The design patterns must support the prevention of safety hazards.* This establishes the alignment with ISO 26262. Security in the vehicle is always aimed at ensuring safety. *R5: The considered approach must contain several design patterns, e.g. in the form of an initial catalog.* We illustrate our evaluation with the following approaches: The approach according to (Fernandez-Buglioni, 2013) describes a method for the application of SDPs and offers an extensive catalog with 80 patterns (R3 and R5 fulfilled). The design patterns contain too many technical details (e.g. technical terms that require a deep understanding of security) for the concept phase (R1 partially fulfilled) and use complex model constructs. that can only be understood by modeling experts (R2 not fulfilled). The patterns focus on IT systems and are rarely related to safety (R4 partially fulfilled). The approach according

https://doi.org/10.1017/pds.2023.268 Published online by Cambridge University Press

To what extent do the appraoches considered satisfy the requirements? Satisfied / Partially s. / Not satisfied		Requirements				
Considered approaches		R1	R2	R3	R4	R5
Anacker et al., 2020	Pattern-based systems engineering	■	■			
Japs, 2021	Resolution of security threats in the system architecture	■	■	■	■	
Amorim et al., 2017	Systematic pattern approach	■		■	■	
Amorim et al., 2020	Safety & security pattern engineering approach	■		■	■	
Cheng et al., 2019	Security patterns for automotive systems	■		■	■	■
Cheng et al., 2020	Security patterns for connected automotive systems	■		■	■	■
Fernandez et al., 2019	Security patterns in practice	■		■	■	■

Figure 1. Rating of existing work.

to (Cheng et al., 2020) describes 10 design patterns for the automotive sector (R5 fulfilled). The design patterns represent countermeasures against security threats (R3 fulfilled) and are safety-relevant (R4 fulfilled). Unfortunately, the descriptions require a deep understanding of security (R1 not met). Furthermore, the complex and detailed UML models (e.g. usage of cardinality, composition, inheritance, deeply nested functions) can only be understood by modeling experts (R2 not met). In particular, the design patterns cannot be applied in workshops with domain experts from different disciplines in the concept phase.

3 INITIAL SECURITY DESIGN PATTERN CATALOGUE

In this section, we present 10 SDPs (cf. Figures 2 and 3). In Section 4 we report on an initial evaluation of our SDPs, based on a student project. To describe the SDPs, we use the SysML diagrams Internal Block Diagram (IBD) and Sequence Diagram (SD). We use SysML because it is the de facto standard modeling language in MBSE (Dori, 2016). IBDs are shown in Figures 2 and 3 on the left and SDs on the right. IBDs are used to describe structural relationships and SDs are used to describe sequences. SysML provides different diagram types for modeling behavior. We chose SDs because the relationship between IBDs and SDs can be communicated in a simple way since the same blocks are used. For simplicity, we omitted stereotypes in this work and use a color scheme to distinguish elements. System elements are shown in blue and elements that interact with the system are shown in yellow.

[01] Authorization *Problem:* The unauthorized access by an unauthorized subject[1] constitutes a security risk. *Solution:* Access to resources is managed through a privilege manager, and these resources are protected from unauthorized access by subjects. The subject's request is either approved or denied after being checked by the privilege manager. *Example:* The privilege manager denies the request if a hacker/subject tries to access the system's protected object for which it does not have permission.

[02] Blacklist & whitelist *Problem:* In certain constellations, very simple protection mechanisms are needed to access a service. For example, if no powerful hardware is available, or there are many users and the manual administration effort must be kept low. *Solution:* In such cases lists can be used. A blacklist prevents access from malicious/untrusted sources. Systems which are whitelisted can be trusted and access is granted. *Example:* Modern vehicles have an infotainment system which allows access to the internet.In a whitelist, the address to a website with software from the vehicle manufacturer can be entered. In a blacklist, known websites which contain harmful programs can be blocked.

[03] Intrusion detection system (IDS) *Problem:* In some scenarios, knowledge about existing attacks from the past can assist in detecting attacks. *Solution:* Usage of a database, which contains known attack patterns. A request to a service is compared to the patterns in the database and rejected if the request is marked as a dangerous attack. If a request is not in the database and leads to an undesired behavior, the request is stored in the database as an attack. *Example:* A component within a network continuously sends slightly changing login requests to another component in order to gain access to a service. Due to the number of changing login requests, a brute force attack is detected as login data is tried to be guessed.

[1] Subject: person or a technical system

https://doi.org/10.1017/pds.2023.268 Published online by Cambridge University Press

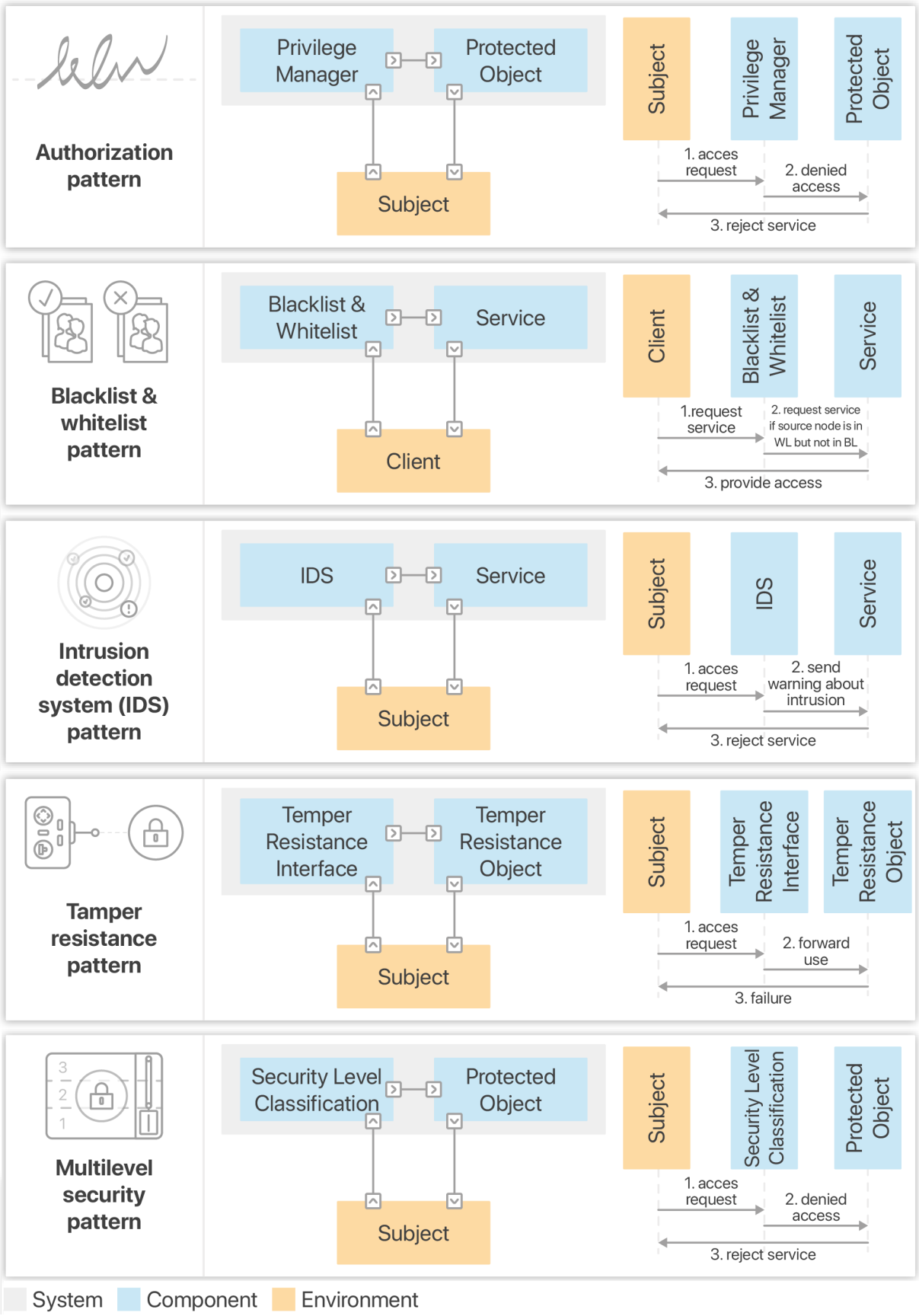

Figure 2. Initial security design pattern catalogue - Part 1

https://doi.org/10.1017/pds.2023.268 Published online by Cambridge University Press

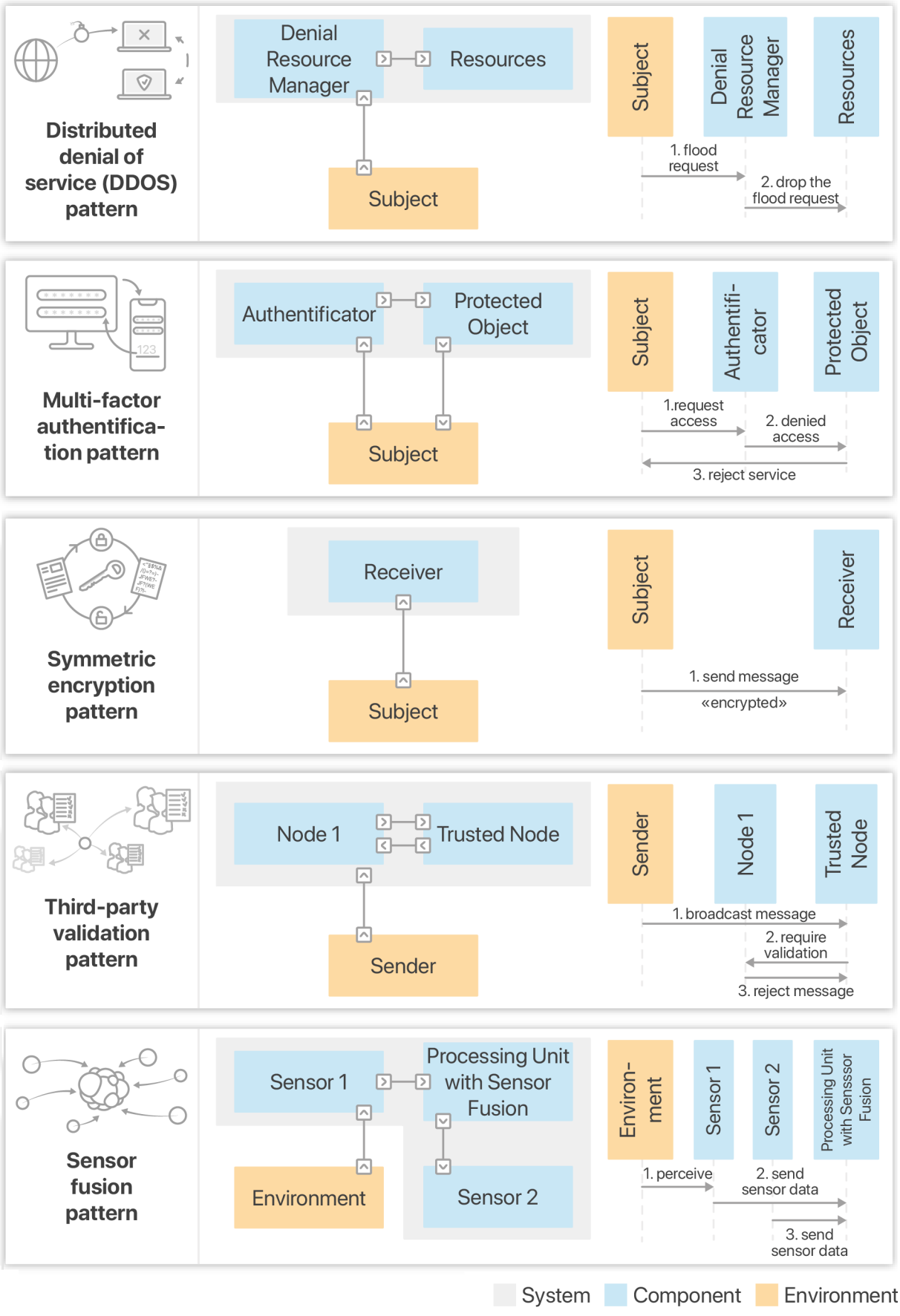

Figure 3. Initial security design pattern catalogue - Part 2

https://doi.org/10.1017/pds.2023.268 Published online by Cambridge University Press

[04] Tamper resistance *Problem:* Unauthorized changes on a system/component lead to vulnerabilities and abnormal system behavior that must be detected, protected, mitigated, or monitored. *Solution:* The Tamper Resistance pattern is simply an interface between the subject and the tamper-resistant object. This interface has a working state that has by default untempered status of the tamper-resistant object. If someone tries to alter the tamper-resistant object then the working state predicts this change that results in breaking the interaction with the tamper-resistant object. *Example:* The vehicle owner wants to install new software via the vehicle's entertainment system (DVD). The software may contain malicious intent that can alter any vehicle component. A tamper resistance pattern prevents these unauthorized component changes by observing its working state resulting in the breaking interaction with the component due to which the software fails to install within the vehicle system.
[05] Multilevel security *Problem:* The problem is about the decision of access in a system with a different security classification. *Solution:* It proposes an access management procedure in a system with the security classification of different levels. Checkpoint contains security level classification for both subject and object. If the subject's security level classification is equal to or greater than the object, access will be granted otherwise denied. *Example:* Messages from external communication interfaces such as telematics systems are assigned to lower trust groups to safe internal ECUs or core systems such as ABS brake system.
[06] Distributed denial of service (DDoS) redundancy *Problem:* DDoS attacks flood malicious requests to network resources resulting in inaccessible service to users that leads to serious consequences. *Solution:* The idea is to provide redundant resources when the particular resource is overburden through service requests. Checkpoint forwards the subject request to the present resource and also notify the Resource Manager about it. The system's redundant Resources are monitored by the Resource Manager and manage with both the Resources and Check Point to balance the loads. *Example:* Vehicles can still communicate if one of the communication channels of connected vehicles is down by switching to other communication resources.
[07] Multi-factor authentication *Problem:* In case that one of the credentials associated with messages/actors within a system is compromised then another authentication level must exist to prevent the system from attacks. *Solution:* The mediator i.e. authenticator interface applies two levels of authentication to get credentials from the protected object at the subject's request. *Example:* A new vehicle wants to join a platoon of autonomous vehicles and only gets admitted after successfully passing two layers of authentication by the network.
[08] Symmetric encryption *Problem:* Important information must not be read by all systems/components as it is safety-critical. *Solution:* The sender encrypts the information through a key to make it cipher, and the receiver decrypts it via the same key to get the correct information. *Example:* A sender is sending a message by encrypting the information using symmetric encryption therefore the attacker (i.e., man in the middle) is unable to manipulate it.
[09] Third-party validation *Problem:* A compromised node in the network may send malicious messages to other network nodes, resulting in unwanted behavior. *Solution:* A sender sends a message to a network of nodes. A receiver trusts the message if at least one other node in the network considers the sender to be trustworthy. There is a trust relationship between the receiver and the other node. *Example:* Two vehicles V1 and V2 drive behind each other on the highway. The rear vehicle V2 trusts the messages from V1. Now V1 and V2 receive a message from a preceding accident vehicle V0. V1 recognizes V0 on the road side in advance and thus trusts the message from V0. Because of V2's trust in V1, V2 also trusts V0's message.
[10] Sensor fusion *Problem:* A single sensor is limited in providing accurate and enough information for the purposes of secure autonomous vehicle driving. *Solution:* Fusions of multiple sensors helped to resolve by compensating the weakness of one sensor with the strength of another providing accurate results that lead to a safe and secure system. *Example:* A camera can be used to detect obstacles on the road e.g. pedestrians but unfortunately do not provide precise depth detection as person imprints in the road environment can be identified as pedestrians. The additional use of a radar sensor increases the quality of object detection by providing in-depth information.

https://doi.org/10.1017/pds.2023.268 Published online by Cambridge University Press

4 EVALUATION

Project characterization: As part of a project at the University of Paderborn, SDPs were applied by 140 master students[2]. The project was part of the Model-Based Systems Engineering (MBSE) course and was designed and managed by us in the context of a research project on automotive systems engineering. The students came from computer science, information systems, and computer engineering backgrounds. The collaboration took place in teams and was conducted virtually. In total, there were 28 teams of 5 people each. The course including the project had a total effort for all students of 25200h. The project lasted 11 weeks, of which 2 weeks were spent on the application of the SDPs. The students spent a total of 8400h on the elaboration of the project, of which 1527h fell on the application of the SDPs. In this work we focus on the evaluation of the application of the SDPs [3]. The project was based on the activities of ISO/SAE 21434 and on existing work on risk analysis using MBSE (Japs, 2020, 2021; Anacker et al., 2021; Anacker and Japs, 2021; Japs et al., 2021) and included the following steps: Teams were formed for preparation. First, we conducted a one-hour competence test. In this test, we asked about the areas of security, safety, requirements engineering and project management, which are relevant for the concept phase of ISO/SAE 21434. Based on this, we proposed team constellations in which the required competencies are represented by at least one person in the team. In addition, there was an option to assemble a team ourselves. In Step 1, use cases for CCAM were first identified (e.g. communication between two vehicles). Based on these use cases, threat cases were identified (e.g. communication failure caused by a jammer). In Step 2, the relevant components (e.g. telematics control unit) and relationships of a system architecture were modeled, which are required to realize the identified cases from Step 1. In Step 3, a risk analysis was performed. The risk analysis consisted of an impact analysis and a feasibility analysis of threat cases. Based on this, decisions on how to handle the risks were defined. In Step 4, if the risks were high, a countermeasure should be selected to minimize the risk. In Step 5, to verify the effectiveness of the countermeasure (e.g. block messages from jammer, Blacklist SDP), a new risk analysis was performed regarding the changed system components and their component relationships. For quality assurance, each team was to evaluate the results of two other teams and provide feedback for improvement. The quality of the evaluations and feedback was the basis on which bonus points were given for the final exam.

Evaluation goal: The project is intended to be a preliminary evaluation of research results, as a basis for evaluation by subject matter experts in the automotive field. We have two evaluation goals: (G1) Quantitative comparison of two variants. In Variant A, countermeasures in the form of the SDPs presented in this paper were allowed to be used. Countermeasures were selected using a selection table. Simplified, this table allows a step-by-step restriction of SDPs based on the following questions: (Q1) Does the SDP fit the system/component level under consideration? (Q2) What kind of protection is needed? In Variant B, countermeasures were to be derived and applied from scientific publications, which are publicly available on the Internet. (G2) Evaluation of the feedback from the teams of both variants.

Quantitative comparison of two test groups: This section refers to Step 4 of the aforementioned procedure in the project. Based on a risk analysis, the teams had to select countermeasures to resolve security-critical vulnerabilities in the architecture. We adopted the procedure for applying countermeasures in the concept phase from (Anacker and Japs, 2021). This procedure served as an application example for the teams. The 28 teams were free to choose the variants. Variant A was chosen by $N_A = 18$ teams and Variant B by $N_B = 10$ teams. To compare the effectiveness of the application of the countermeasures, each team had to record several indicators. In the following, we present the evaluation on the most important indicators regarding the application of SDPs.

Based on the use cases and threat cases from Step 1, models were created in Step 2 in the form of an IBD (system architecture) and several SDs (system behavior).

Functions are part of the SDs and represent the relationships between individual components in the IBD. Figure 4 compares the number of functions created by the teams of the two variants. A function is triggered in one component/system and has an effect in the same component/system or in another component/system. The application of countermeasures affects a subset of the functions created. On median,

[2] A total of 175 students started the project, of which 140 students completed it.

[3] In this work we only processed results from students from whom we received an agreement. This was the case for almost all students. In particular, we did not penalize any student for not agreeing.

https://doi.org/10.1017/pds.2023.268 Published online by Cambridge University Press

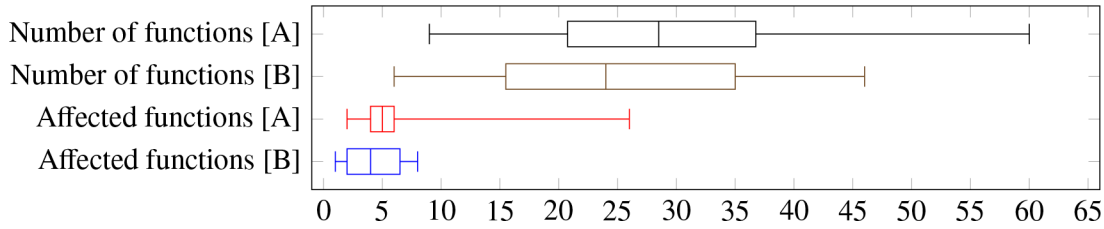

Figure 4. Affected functions by applying the countermeasure, $N_A = 18$ Teams, $N_B = 10$ Teams

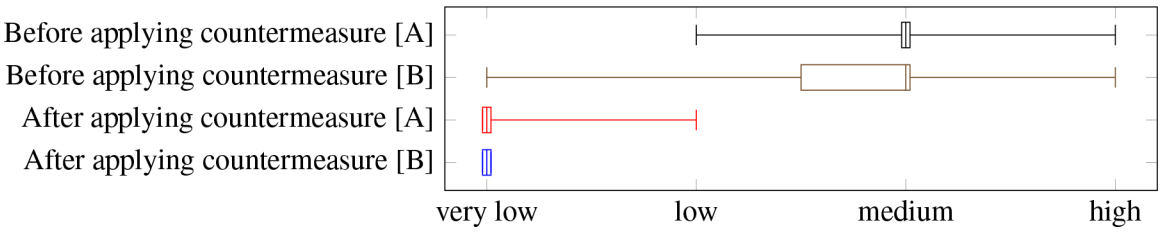

Figure 5. Assessing the feasibility of hacking attacks, $N_A = 18$ Teams, $N_B = 10$ Teams

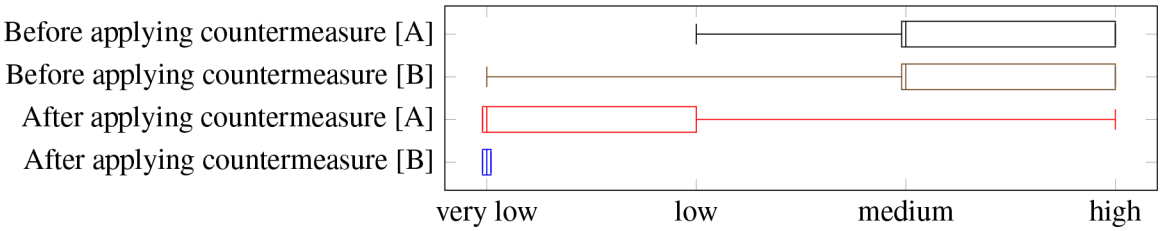

Figure 6. Assessing the safety impact of hacking attacks, $N_A = 18$ Teams, $N_B = 10$ Teams

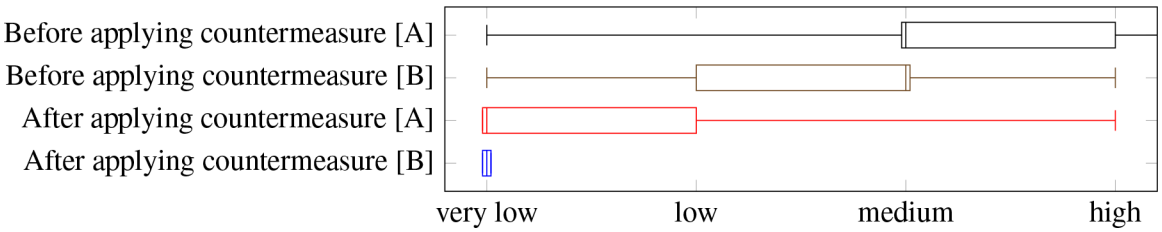

Figure 7. Calculated risks, $N_A = 18$ Teams, $N_B = 10$ Teams

Variant A teams created 28 functions, of which 5 were affected by the countermeasure in median. For Variant B, there were 24 functions of which 4 were affected by the countermeasure. The median is almost identical for both variants before and after application of the countermeasure. The box indicates the range in which 50% of the data is located. In this case, there is almost no difference for both variants. *Interpretation:* The selection of the variant had no clear influence on the number of functions created or the number of functions affected by the countermeasure. ISO/SAE 21434 requires the creation of a feasibility rating and an impact analysis to calculate the risk. Figure 5 shows the feasibility rating data for the teams of both variants before and after applying the countermeasures. Respectively, the safety impact analysis data and calculated risks are shown in Figures 6 and 7. The median of the feasibility rating, impact analysis, and calculated risks is mostly identical for both variants before and after the countermeasure was applied. For the majority of the feasibility rating data, the values are identical before and after applying the countermeasure. *Interpretation:* The selection of the variant did not have a clear influence on the feasibility rating. The safety impact is identical for both variants for 50% of the data. After applying the countermeasure, 50% of the data are in the very low/low range for both variants. *Interpretation:* The safety impact after application of the countermeasure is very low for Variant B. This could be due to the fact that the number of 10 SDPs for Variant A was too low. In contrast, the number

https://doi.org/10.1017/pds.2023.268 Published online by Cambridge University Press

of available publications in Variant B, which present countermeasures, is much higher. As a result, more suitable countermeasures could be selected in Variant B, which led to a lower safety impact. In order for Variant A to be competitive with Variant B, the initial design pattern catalog must be expanded. The calculated risk differs more strongly for both variants. Basically, there is a high or medium risk for Variant A for 50% of the data, which shifts to low or very low after applying the countermeasure. Similarly, for Variant B, for 50% of the data, the risk shifts from medium or low to very low. The risk depends on the feasibility rating and the safety impact. The values of the feasibility rating are almost the same for both variants. The data for Variant B basically has a lower value with regard to the safety impact. As a result, the risk for Variant B is generally lower.

	…was difficult	… was helpful
Both	D1: Choice between different possible countermeasures. (3x)	H1: Understanding of own architecture based on Step 3 to apply countermeasure. (13x) H2: Provided application example. (7x)
B: Internet	D2: High effort to understand the sources found and high effort to apply the countermeasure found. (9x)	H3: Abstraction of the descriptions and models of the countermeasures found. (1x)
	D3: High effort to find the appropriate countermeasures. E.g. Initially the descriptions seem to fit. On closer examination, the countermeasures did not fit. (6x)	
A: Pattern	D4: Transition of the design pattern to self-created architecture. (4x) D5: Use of the selection table did not limit countermeasures to exactly one countermeasure. (5x)	H4: Description and models of design patterns. E.g. by comparing the SDP descriptions or by checking which elements of the SDP fit to the own architecture. (11x)
	D6: The description of the design patterns was not sufficient for a deep technical understanding. (1x)	H5: Basically, the selection table was helpful to narrow down the possible countermeasures. (9x)

Figure 8. Aggregation of the feedback from the 28 teams.

Evaluation of feedback In the following, we summarize the feedback of the 28 teams and derive the next steps from it [4]. The positive feedback H1-H5 is self-explanatory. Regarding D1, D4, D5, D6: We believe that the reference to detailed information in the form of scientific publications as a supplement to the SDPs will improve this. Regarding D2, D3: The work in the concept phase is characterized by the collaboration of leading subject matter experts who often have little time (Japs, 2021). Variant B is not suitable for use in such workshops, because choosing freely from very many possible countermeasures described in great detail requires a lot of time for understanding and application. An enlargement of the pattern catalog could help here.

5 SUMMARY AND FUTURE WORK

The development of modern vehicles is complex, especially with regard to compliance with security and safety. ISO/SAE 21434 considers security and safety along the entire product life cycle. According to the standard, a system architecture, a risk analysis and the application of countermeasures are carried out in the early system design. Design patterns are solutions to known design problems. Security Design Patterns (SDP) describe countermeasures and are used to reduce risk. After our literature review, we did not find a suitable approach that presents SDPs that would be applicable in early system design. In this paper, we presented 10 SDPs for early system design, which we evaluated during an 11-week student project with 28 teams. In Variant A, 18 teams used SDPs as countermeasures to reduce risk. While in Variant B, 10 teams chose to use countermeasures from scientific publications. We quantitatively evaluated the results of all 28 teams with respect to several indicators. Here, we could not find any significant difference quantitatively. The number of diagrams and functions, the security and safety ratings, and the resulting risk of both variants were almost the same. We also evaluated the feedback from all teams. Particularly noticeable were the following points. The teams in Variant B often mentioned the high effort required to find suitable countermeasures (6 out of 10 teams). Furthermore, in Variant B, 9

[4] In total, we received feedback in the amount of 8 DIN A4 pages. For high quality positive and negative descriptions the teams could get bonus points for the final exam

https://doi.org/10.1017/pds.2023.268 Published online by Cambridge University Press

out of 10 teams stated that it was very time consuming to understand and apply the countermeasures in the scientific publications they found. We suspect that the Variant B approach is not attractive enough for repeated use of risk analysis when applying multiple countermeasures in succession because of the higher time required. SDPs are the better choice in this context. In future, we want to define further indicators that will allow us to determine the differences between the two variants more precisely. Furthermore, the approach is to be carried out with subject matter experts from the automotive industry in order to get further suggestions for improvement regarding the description and design of the SDPs.

ACKNOWLEDGMENTS

This research was funded by the German Federal Ministry of Education and Research (BMBF) in the project SecForCARs, grant number 16KIS0790. The contents of this publication are the sole responsibility of the authors. We thank Oliver von HeiÃ§en for his support in determining bonus points.

REFERENCES

Amorim et al. (2017), "Systematic pattern approach for safety and security co-engineering in the automotive domain", In: International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2017), Trento, Italy, http://doi.org/10.1007/978-3-319-66266-4_22.

Anacker, H. et al. (2020), "Pattern based systems engineering - Application of solution patterns in the design of intelligent technical systems", In: 16th International design conference, Cavat, Dubrovnik, Croatia, http://doi.org/10.1017/dsd.2020.107.

Anacker, H., Japs, S. (2021), "Resolution of safety relevant security threats in the system architecture design phase on the example of automotive industry", In: Proceedings of the design society, 1, http://doi.org/10.1017/pds.2021.517.

Anacker, H., Dumitrescu, R., Japs, S. (2021), "SAVE: Security & safety by model-based systems engineering on the example of automotive industry", In: Procedia CIRP, vol. 100, 187–192, http://doi.org/10.1016/j.procir.2021.05.053.

Cheng et al. (2019), "Security patterns for automotive systems", In: ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS-C), Munich, Germany, http://doi.org/10.1109/MODELS-C.2019.00014.

Cheng, B. et al. (2020), "Security Patterns for Connected and Automated Automotive Systems", In: Journal of Automotive Software Engineering, http://doi.org/10.2991/jase.d.200826.001.

Dori, D. (2016), "Model-Based Systems Engineering with OPM and SysML", Springer.

Fernandez-Buglioni, E. (2013), "Security Patterns in Practice: Designing Secure Architectures Using Software Patterns", Wiley.

Gausemeier, J., Rammig, F.J., Schäfer, W. (2014), "Design methodology for intelligent technical systems", Springer, Berlin-Heidelberg, http://doi.org/10.1007/978-3-642-45435-6.

Husung, S. et al. (2021), "Using model-based systems engineering for need-based and consistent support of the design process", In: Proceedings of the Design Society, 1, http://doi.org/10.1017/pds.2021.598.

Japs, S.(2020), "Security & safety by model-based requirements engineering", In: IEEE 28th International Requirements Engineering Conference (RE), 422–427. http://doi.org/10.1109/RE48521.2020.00062.

Japs, S. (2021), "Towards the development of the cybersecurity concept according to ISO/SAE 21434 using model-based systems engineering", In: IEEE 29th International Requirements Engineering Conference (RE), 486–491. http://doi.org/10.1109/RE51729.2021.00073.

Japs, S. et al. (2021), "D-REQs: Determination of security & safety requirements in workshops based on the use of model-based systems engineering", In: IEEE 29th International Requirements Engineering Conference Workshops (REW), 412–414. http://doi.org/10.1109/REW53955.2021.00073.

Martin et al. (2020), "Combined automotive safety and security pattern engineering approach", In: Reliability Engineering & System Safety, Volume 198, http://doi.org/10.1016/j.ress.2019.106773.

TÜV Thüringen (2022), "ISO/SAE 21434 – Standard zur Cybersecurity im Automobilbereich", https://tuev-thueringen.de/blog/iso-sae-21434-standard-zur-cybersecurity-im-automobilbereich, last access: 2022-11-28.