Limitation and Improvement of STPA-Sec for Safety and Security Co-analysis

Christoph Schmittner[1], Zhendong Ma[1], and Peter Puschner[2]

[1] Department of Digital Safety & Security
AIT Austrian Institute of Technology
christoph.schmittner.fl@ait.ac.at; zhendong.ma@ait.ac.at
[2] Vienna University of Technology,
Department of Computer Engineering
peter@vmars.tuwien.ac.at

Abstract. Safety-critical Cyber-physical Systems (CPS) in vehicles are becoming more and more complex and interconnected. There is a pressing need for holistic approaches for safety and security analysis to address the challenges. System-Theoretic Process Analysis (STPA) is a top-down safety hazard analysis method, based on systems theory especially aimed at such systems. In contrast to established approaches, hazards are treated as a control problem rather than a reliability problem. STPA-Sec extends this approach to also include security analysis. However, when we applied STPA-Sec to real world use cases for joint safety and security analysis, a Battery Management System for a hybrid vehicle, we observed several limitations of the security extension. We propose improvements to address these limitations for a combined safety and security analysis. Our improvements lead to a better identification of high level security scenarios. We evaluate the feasibility of the improved co-analysis method in a self-optimizing battery management system. We also discuss the general applicability of STPA-Sec to high level safety and security analysis and the relation to automotive cybersecurity standards.

Keywords: Cyber-physical Systems, safety and security co-analysis, STAMP, STPA-Sec, automotive cybersecurity

1 Introduction

Safety-critical cyber-physical systems (CPS) become increasingly complex and interconnected. For example, the future transportation system is envisioned to be intelligent and interconnected, in which heterogeneous Information and communications technologies (ICT) and physical elements, from vehicular systems (e.g. hybrid vehicles) to energy provider and infrastructure components, interact with each other and the physical environment to be self-organizing for an optimized multi-modal mobility strategy for drivers, passengers, and goods. Such an interconnection introduces cybersecurity risks which might threaten the safety of the system. As a results, system analysis must consider safety and security in order to define system goals and a concept for a safe and secure system.

One of the challenges is to identify potential safety and security risks at the beginning of a system's development lifecycle where only high level information is available for the analysis (i.e. the "concept phase"), in order to reduce and mitigate the risks to an acceptable level. As one of the new approaches, STPA-Sec[1] is an extension of the System-Theoretic Process Analysis (STPA)[2, 3] that extends the safety analysis method with security considerations. In STAMP, systems are modeled as hierarchical structures in which higher level controllers control processes at lower levels via actors. The lower levels send feedback to the higher levels via sensors. The output of STPA-Sec is a list of system-level scenarios which can lead to losses. In this paper, we identify several limitations of STPA-Sec and propose improvements when applying STPA-Sec for safety and security co-analysis. Specifically, we improve the annotated control loop used in STPA for causal analysis for identifying unsafe control actions due to security attacks. We evaluate and demonstrate the improved STPA-Sec by applying it to a connected and self-optimizing battery-management system (BMS) for hybrid vehicles.

In the following, Section 2 gives an overview of the State of the Art, Section 3 introduces and discusses STAMP (Systems-Theoretic Accident Model and Processes), STPA and STPA for Security (STPA-Sec). Section 4 presents our improvements to STPA-Sec for safety and security co-analysis. Section 5 applies the improved STPA-Sec to the case studies and a real-world scenario. Section 5.3 discuss the general applicability of STPA-Sec to safety and security co-analysis and its relation to automotive cybersecurity standards, followed by the conclusion in Section 6.

2 State of the Art

The automotive safety standard ISO 26262 [4] divides the system lifecycle into Concept, Development (System, Hardware, Software) and Productions, Operation and Maintenance phase. In each phase specific activities and work results are defined. The main goal during the concept phase is to define functional safety requirements and functional safety concept. Both are based on the safety goals which result from the hazard and risk analysis (HARA). During HARA, hazards and risks are identified and rated. Based on this rating, an automotive safety integrity level (ASIL) is defined, which denotes the required risk reduction. Recently the first automotive cybersecurity standard SAE J3061[5] was published, proposing a security engineering process in parallel or joint with the safety lifecycle. Regarding the application of a cybersecurity process in conjunction with a safety process tailored to ISO 26262, the standard proposes the following:

> The integration of these activities may be done by keeping the Cybersecurity and safety activities separate, but performing these activities in conjunction with each other and with the same team, or parallel activities may be done by developing an integrated technique that covers both safety and Cybersecurity at the same time. An example of this is to

> develop a technique to perform both a hazard analysis and risk assessment, and a threat analysis and risk assessment at the same time using a single integrated template and method. A tightly integrated process for Cybersecurity and safety has the advantage of a common resource set, thus, requiring fewer additional resources. [5]

In recent years, multiple methods for safety and security co-analysis have been developed, aiming at a combined approach towards safety and security. SAHARA (A Security-Aware Hazard and Risk Analysis Method) [6] extends the classical Hazard and Risk analysis with security related guide words and an evaluation of risks. FMVEA (Failure Mode, Vulnerabilities and Effects Analysis) [7] extends the Failure mode and effects analysis with threat modes and vulnerabilities. CHASSIS (Combined Harm Assessment of Safety and Security for Information Systems) [8] is a methodology for safety and security assessments and formulation of mitigation measures, based on use case and sequence diagram modeling. Other approaches focus less on the identification of security related hazards but more on a detailed analysis via extended Fault Trees [9–11].
In this paper, we will investigate STPA-SEC[1], a top-down safety and security analysis method, to the concept phase of an automotive use case and evaluated the results according to the requirements provided by ISO 26262 and SAE J3061.

3 Review of STPA-Sec

System-theoretic Process Analysis for Security (STPA-Sec) [1] extends the safety-focused System-theoretic Process Analysis (STPA) method for security analysis. Both methods are based on the theory of STAMP (Systems-Theoretic Accident Model and Processes) [2]. In STAMP, systems are modeled as hierarchical structures in which higher level controllers control processes at lower levels via actors. The lower levels send feedback to the higher levels via sensors. STAMP views safety accidents as a result of a lack of control, instead of a chain or sequence of events. Since modern systems are increasingly complex with multiple interacting elements, it is difficult to identify root causes for accidents. STPA-Sec examines each control action under different possible conditions and guide words and identifies loss scenarios. Losses are interpreted as insufficient or missing controls or safety constraints.
Step 1: Establishing the systems engineering foundation. STPA-Sec takes a top-down approach focusing on identifying unacceptable losses and vulnerable states in order to locate essential system services and functions to be protected and controlled. The first step identifies such unacceptable losses.
Step 2: Creating a model of the high level control structure. In this step the control model of the system is generated. A control model consists of the controlled processes , controller, sensors and actors and relevant control actions and sensed process variables.
Step 3: Identifying unsafe/insecure control actions. All control actions

from the control model are reviewed. Unsafe or unsecure control actions or control actions leading to hazards (or vulnerable system states) are identified, based on four guide phrases (i.e. “control action not given”, “control action given incorrectly”, “wrong timing or order of control action”, and “control action stopped too soon or applied too long”).

Step 4: Developing security requirements and constraints/identifying causal scenarios. In this step, unsafe/insecure control actions are extended to unsafe/insecure scenarios in order to identify missing high level safety/security constraints. An unsafe/insecure scenario consists of an unsafe/insecure control action, context and potential causes. Intentional causal scenarios, e.g attack scenarios, are identified by analyzing the physical and logical infrastructure similar to established security analysis. An annotated control graph (cf. Fig. 1) supports the identification of potential causes for unsafe control action.

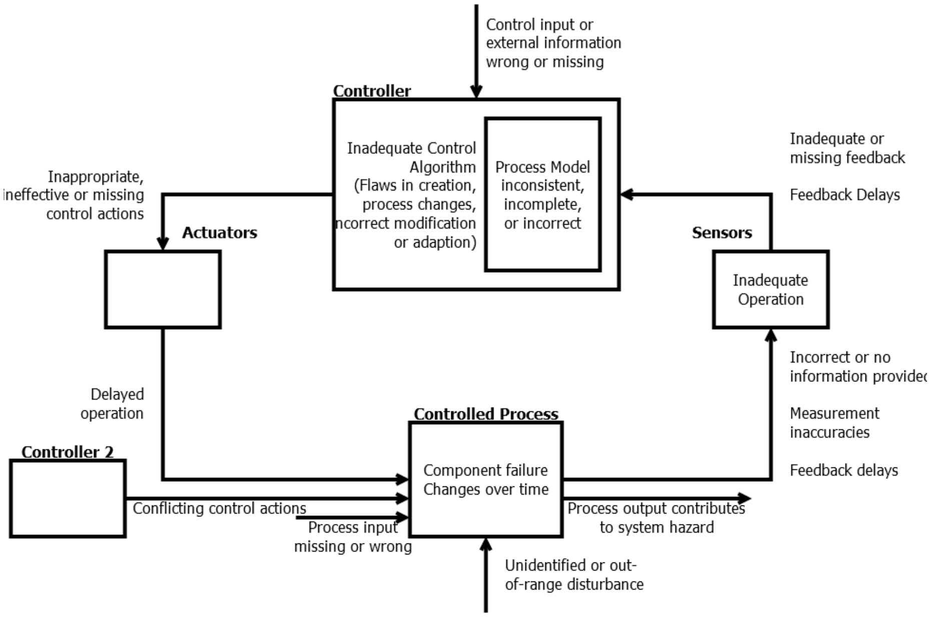

Fig. 1. Control loop annotated with potential starting points for the identification of unintentional causes for unsafe control actions [3]

During our application of STPA-Sec, we identified two limitations in the co-analysis of safety and security. First, guidance for the identification of intentional causal scenarios is challenging to apply. While some terms in Fig. 1 can be interpreted in an intentional and security related way (e.g.“incorrect or no information provided” could be interpreted as result of a Denial of Service), we found it helpful to explicitly include such guidance. Second, another restriction of the control loop model is the exclusion of security relevant elements. The control loop describes the control process for the system in the intended configuration. However, attackers, undesired influences from elements outside the intended control model, and the exchange of sensitive information not relevant for the control model are not included. While the control model is an important view on the system and its behavior, it does not adequately capture the view of a potential attacker.

4 Extension of STPA-Sec

We propose the following extensions to improve the aforementioned limitations.

4.1 Alignment of Terminology

The safety and security community have developed their own terminologies. Safety and security co-analysis and interaction require clear and correct communication between the two communities. Some of the STPA-Sec terms are safety-oriented, leading to ambiguity and misunderstanding for security analysis. The problem is aggravated because some terms in STPA even deviate from standard safety terminology.

We address this issue by aligning important terms used in STPA-Sec in a safety and security context. First, we identify the terms in STPA-Sec that cause ambiguities. When possible, we use STPA terms as anchors and align the security terms to them. Second, we provide definitions for these terms that are valid for both safety and security. Third, we resolve potential differences between safety and security terminology, and add security-oriented terms that are necessary for co-analysis. We also add definitions to terms in STPA-Sec that are different from common safety terminology. We base our security definition on [12], commonly accepted in the security community. Table 1 shows the resulting terminology. Note that our intention is not to define a comprehensive vocabulary, but rather to establish a common understanding, helpful for safety and security co-analysis.

4.2 Guidance for the Elicitation of Intentional Scenarios

In order to improve the guidance for the identification of intentional unsafe control scenarios (e.g. attack scenarios leading to the activation of an unsafe control action), we propose an extension to the security approach in STPA. The control model is reviewed by a team of experts for potential causal scenarios for the intended and malicious activation of unsafe control actions. We extend the annotated control graph as a starting point for the investigation of intentional scenarios. This guidance should not be seen as a checklist which covers all possibilities. It is intended as a starting point for further thoughts and the investigation of unsafe and insecure scenarios.

Fig. 2 shows the classic guidance for the investigation of unintentional scenarios and, in red and cursive, our extensions for the investigation of intentional scenarios. In addition to the annotations, we add a spoofed controller in order to note the potential that conflicting control actions are intentionally introduced in the system. It is important to consider external unplanned interactions during the investigation of potential causes.

Table 1. Safety and security terminology

Terms	Definition
attack	attempt to gain unauthorized access to or make unauthorized use of an asset
accident	event which causes undesired losses of life, availability etc.
control	in general, alter the operation condition of a system; in security, measure that is modifying risk
control loop	model describing the control flow of a system or process. The model consists of one or more controllers, controlled processes, sensors and actuators
event	occurrence or change of a particular set of circumstances, also refers to an incident or accident
hazard	dangerous system states which can lead to accidents.
threat	potential cause of an unwanted incident, which may result in harm to a system
unsafe control action	control action which can cause, under certain circumstances, hazards.
unsafe control scenario	scenario which describes context and potential causes for the execution of an unsafe control action.
loss event	accident
vulnerability	vulnerable system state, STPA-Sec uses it to refer to hazard; in security, weakness of an asset or control that can be exploited by one or more threats

5 Case Study

We evaluate our extended approach by reviewing a number of already identified scenarios in order to backtest if these scenarios are identifiable with the improved guidance for identifying intentional scenarios. In addition, we demonstrate the application to a real-world complex and connected automotive system.

5.1 Backtest of Existing Scenarios

The first reviewed scenario is from the automotive domain. Miller and Valasek identified the missing authentication of control actions in automotive on-board networks as a major vulnerability [13]. If there is no control over the source of a control action, the safety and security depend completely on the insulation of the control system. Guaranteeing that a system is completely free from undesired control actions is much harder than restricting the allowed sources for control actions. Injecting manufactured control actions in an otherwise unaltered system will cause conflicts between legitimate and illegitimate control actions and the consequences and system reactions are not predictable. Such intentional scenarios for unsafe control actions are identifiable with the new guidance *sending manufactured control actions/input, overriding legitimate control actions/input.* Kundur *et al.* [14] presented a typical cyber attack scenario for smart grids.

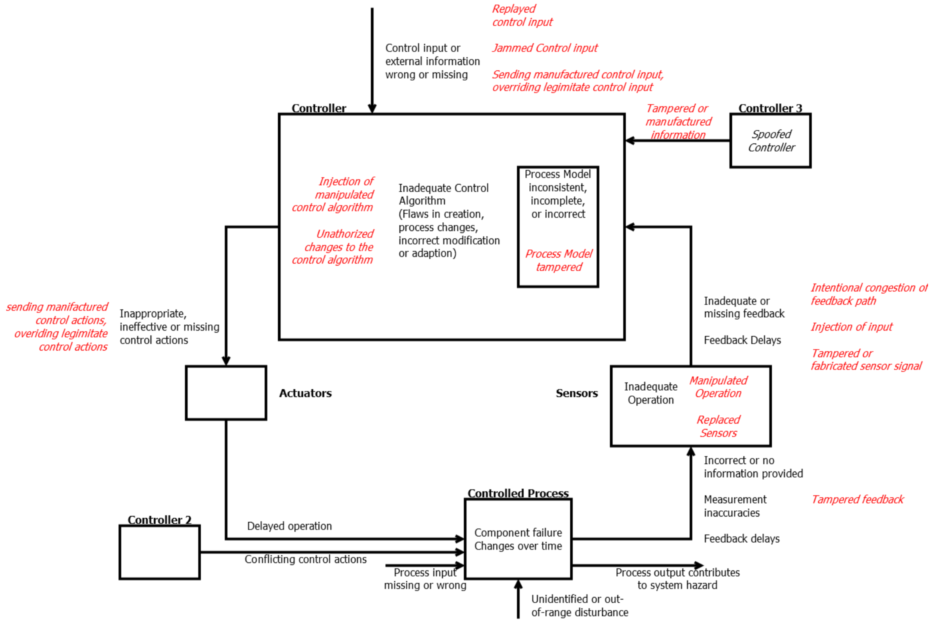

Fig. 2. Control loop with starting points for the identification of malicious scenarios

The scenario includes fabricating or tampering sensor information which causes incorrect decision for the controller regarding necessary control actions for the load management. Such incorrect decisions or unsafe control actions could result in generator trip out. This scenario and similar are covered with the extended guidance for the control path between sensor and controller, especially with *Tampered of fabricated sensor signal*, in the extended annotated control graph from figure 2.

Dadras *et al.* [15] presented an interesting attack scenario on vehicular platooning. An adversary vehicle is able to exploit the control logic and destabilize the platoon or influence position and velocity of other vehicles through local changes. By accelerating and braking in a certain frequency range they caused the control algorithm of other vehicles to become instable and oscillating. This manipulation is aimed at the sensor input and exploits known weaknesses in the control algorithm to cause unsafe control algorithm. Such attacks are difficult to perform and require knowledge of the internal workings of the attacked system. Since they are also very difficult to detect and defend against there is some interest to explore such threats. Krotofil *et al.* [16] demonstrated similar behavior in an industrial context, the process in a chemical plant was attackable by manipulating the input of a few sensors. Such scenarios are identifiable with the extend guidance for considering *tampered feedback* from the controlled process.

Although not comprehensive, the backtesting of existing scenarios shows that our approach is better at identifying unsafe and unsecure scenarios using limited and high level information in the system development concept phase.

5.2 Analysis of Battery Management System

We apply our extended STPA-Sec approach to the analysis of a Battery Management System (BMS) for hybrid vehicles[6]. A BMS optimizes the driving strategy, e.g. usage of the electrical engine or combustion engine, based on multiple factors (including the goal for the driving experience chosen by the driver, e.g. minimizing energy consumption, maximizing acceleration and external factors like chosen route, charging opportunities, temperature). Besides, the BMS also directly controls the high-voltage battery system that provides power supply for the electrical engine, and charges the battery from external or internal sources. The different charging scenarios include charging from an external source, the plug-in charging, or charging from internal sources, regenerative braking and the usage of the on-board (internal) combustion engine as generator.

Step 1 - System Description and Hazard and Accident Identification Fig. 3 gives an overview of the architecture of the BMS and the on-board network. The BMS estimates the state of the battery system by monitoring total and cell voltage, cell temperature and current. Based on the measured values and an internal model of the battery behavior, it calculates the state of charge and state of function(health). It controls discharging and charging of the battery and ensures the safe operation, ensured by restricting the system to its safe operating area. The BMS can partially control the environment (heating or cooling) and request a restricted usage of the battery by sending a message via the CAN-Bus to other control units. The complete high-voltage system can also be de-energized. The CAN communication is utilized for the communication with other control units and to receive information about external parameters which influence the control strategy. External communication and remote connectivity is accessed via the Telematic Unit. In addition, the BMS is directly connected to the outside via the charging interface [17].

Fig. 4 shows the control model for the Battery Management System. This control model is the result of an iterative process of extending and refining. We started with a rather simple model of the battery as a controlled process and the BMS as controller and refined it.

The low level BMS estimates the state of charge, based on the internal process model of the battery, e.g. how voltage and current relate to the overall charge in the battery. It influences the energy consumption of the vehicle and maintains safe operation of the battery. This is done by controlling the environment and restricting or influencing the discharge of the battery. It is responsible for the short term control strategy for the battery.
The high level BMS is responsible for the long term control strategy for the

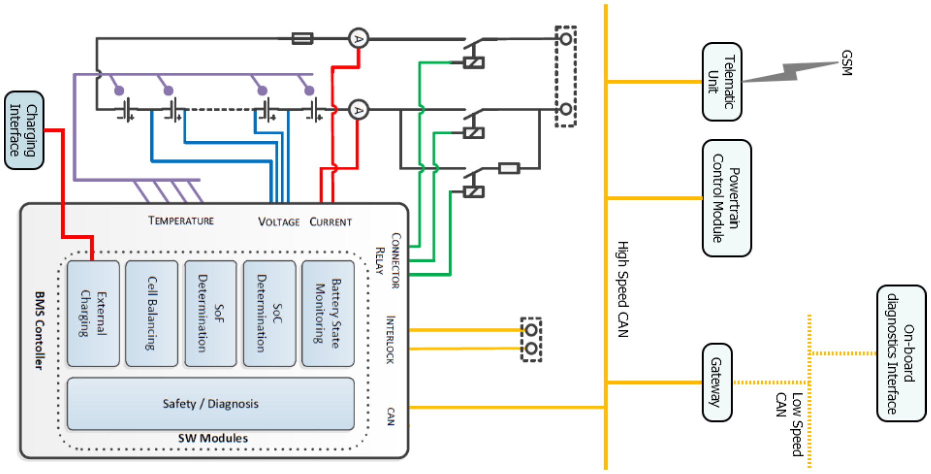

Fig. 3. Architecture of the Battery Management System [6]

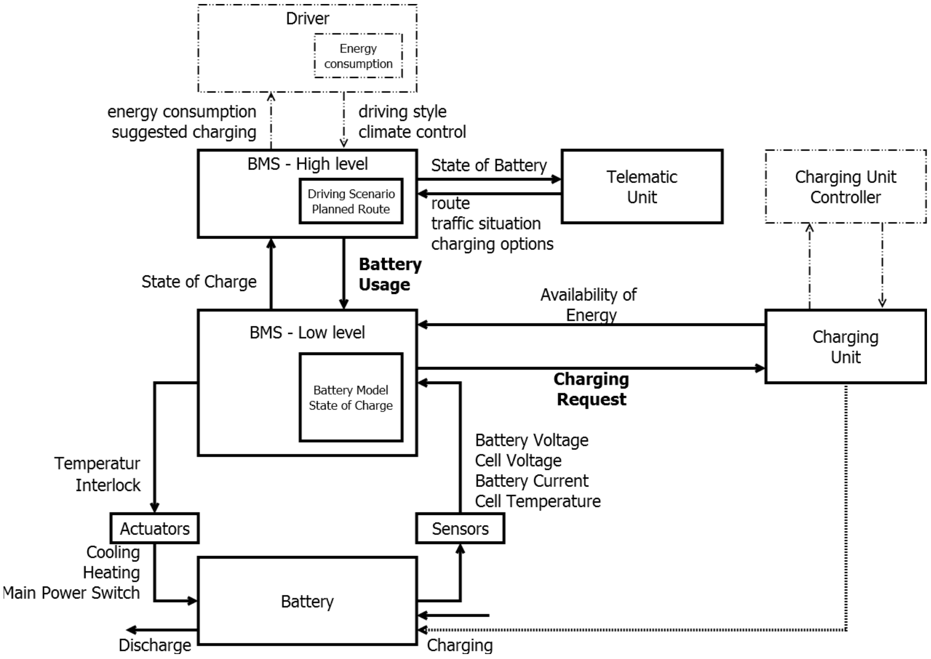

Fig. 4. Control Loop of the Battery Management System

battery. The goal is to optimize the battery usage. This system relies on the state-of-charge and state-of-health data from the low level controller and combines this information with external data about charging options, costs, route and traffic situation. It interconnects with other vehicles and the infrastructure and optimizes the driving strategy for each vehicle based on the overall traffic flow and energy consumption.

While the driver is not directly involved in controlling the battery, most hybrid vehicles tend to give feedback to the driver on whether he or she is driving ecologically [18]. In some vehicles this information is also available via mobile apps [19]. These apps also allow some level of remote control of the Battery Management System.

In Table 2, we list some of the possible hazardous scenarios for the battery management system.

ID	Hazard	Comment
H1	over current	different for charging or discharging
H2	over voltage	only relevant during charging
H3	under voltage	may permanently damage battery performance
H4	over temperature	may damage battery
H5	under temperature	temporary reduces battery performance
H6	over pressure	may damage battery
H7	Ground fault or leakage current	undesired flow of power
H8	Reduced Control over vehicle speed	undesired acceleration or deceleration

Table 2. Table of identified hazards

Based on the list of identified hazards, the following accidents are related to the BMS (Table 3). We focused on safety related loss events and excluded other losses like financial or operational losses.

ID	Accident	Related Hazards
A1	Electric Shock for vehicle passengers or persons touching the vehicle	H7
A2	Battery causes a vehicle fire	H1, H2, H4, H6
A3	Collision with object or other vehicle	H1, H2, H4, H6, H8

Table 3. Table of identified accidents

Step 2 - Identification of Unsafe Control Action Due to the high number of unsafe control actions generated by STPA-Sec we present only an excerpt of all identified unsafe control actions. In order to identify unsafe control actions all control action from the control loop model of the system 4 are reviewed, using the guide phrases3. Identified unsafe control actions are then linked to the hazards.

Control Action: Charging Request

1. Control Action not given
   (a) Battery is not charged (non-hazardous)
2. Control Action given incorrectly

   (a) Excessive charging request[3] is transmitted to charging unit during plug-in charging (H1, H2, H4, H6)

3. Wrong timing or order of Control Action
   (a) Charging request to charging unit for plug-in charging is transmitted before battery system is ready to charge (H4)
   (b) Charging curve is transmitted to charging unit in wrong order for plug-in charging, constant current and constant voltage phases are swapped (H1, H2, H4, H6)
4. Control Action stopped too soon or applied too long
   (a) Charging Request to charging unit for plug-in charging is transmitted after battery has been fully charged (H1, H2, H4, H6)
   (b) Transmission of charging request to charging unit for plug-in charging has been stopped too soon (non-hazardous)

Control Action: Battery usage

1. Control Action not given
   (a) Battery usage strategy is not transmitted to low-level BMS, State of charge or driving strategy is not considered for the discharge of the battery (H3, H8)
2. Control Action given incorrectly
   (a) Command given to charge battery during driving by recuperative braking while battery is fully charged (H1, H2, H4, H8)
   (b) Command given to charge Battery while vehicle is driving in electric mode (H8)
   (c) Command given to use battery for electrical engine is given while battery charge is critically low. (H3)
3. Wrong timing or order of Control Action
   (a) Command to charge battery by combustion engine is given during plug-in charging (H1, H2, H4)
4. Control Action stopped too soon or applied too long
   (a) Battery usage requested while Battery voltage is critically low (H3)

Step 3 - Identification of Intentional and Unintentional Scenarios for Unsafe Control Actions Based on the extend annotated control loop (cf. Fig. 2), we identified intentional and unintentional causal scenarios for the unsafe control actions. The identification of the causal scenarios is done via an review of the unsafe control actions, the control model and the annotated control model by experts from the safety, security and automotive domain.

Excessive charging request is transmitted to charging unit (H1, H2, H4, H6)

[3] Depending on the phase in the charging cycle and the battery there are limits to voltage and current which, when exceeded, may damage the battery.

- An excessive charging request can be caused by a modified charging request from the BMS to the charging unit due to tampered process model in the BMS software to enable fast charging for non-fast chargeable batteries. Potential motivation for the owner is that he is interested in faster charging and does not care about longevity of battery due to leasing contract for battery.
- A wrong charging request from BMS to charging unit may be caused by a failure/design error in the temperature sensor for a battery. Due to financial reasons a malicious manufacturer could reduce the number of sensors per battery cells below the number required for a reliable reading.
- Even when the vehicle BMS requests the correct power level a manipulation on the communication between BMS and charging unit could lead to an unsafe charging request. Such a manipulation could be directed at the charging unit or the central charging management system at the backend.

Command given to use battery for electrical engine is given while battery charge is critically low. (H3, H8)

- Battery usage strategy is "optimized" and replayed battery messages are injected into the CAN Bus via the On-board diagnostics interface while the battery charge is critically low. This increases vehicle performance while decreasing life time of the battery and may cause permanent damage to the battery or even the vehicle. Owner is interested in maximized battery usage and does not care about longevity of battery due to leasing contract for battery.
- The battery is replaced with a different model with a changed behavior. The corresponding process model in the controller is not changed which leads to a mismatch between physical process and assumed process behavior. A mechanic shop could do this in order to save money or because they have no access to the internal process models in the controller.
- A compromised telematic unit could be used to send messages which impersonate the high level BMS. This is easy in a CAN Network since messages only carry a receiver ID and no sender ID. Therefore any compromised Electronic Control Unit (ECU) could be used to send commands which trigger unsafe control actions. The telematic unit is especially vulnerable, because most of the external communication is done via this ECU.

5.3 Evaluation and Discussion

STPA-SEC is strongly focused on the considered and intended control model to identify deviations. While this is sound for the identification of safety related effects, it does not cover more information-security centric considerations such as privacy, as recommended in SAE J3061 to apply cybersecurity engineering to all elements including Personally Identifiable Information (PII). Hence losses related to privacy and confidentiality are currently not considered in STPA-Sec or in the control model. STPA-Sec excludes the flow and exchange of data not directly connected to the control of a process, e.g. battery usage information collected for insurance reasons. Connections not directly related to the control

flow but can be misused might be difficult to identify within STPA-Sec. An analysis based on an architectural or dataflow model would be better suited for the identification of such risk scenarios.
In addition the general approach of STPA, applying the four guide phrases to all control actions from the control loop model of the system to identify multiple unsafe control actions and then developing multiple unsafe scenarios leads to a very high number of unsafe scenarios which need a strict filtering.

With respect to the automotive cybersecurity guideline SAE J3061 [5], we have the following opinions. The list is divided according to the lifecycle steps in SAE J3061, which are modeled after the lifecycle presented in ISO 26262 [4].
Item Definition Step 1 in STPA-Sec requires similar information about the system as the item definition. There is a stronger focus on potential interactions and interfaces with systems outside the control model in the existing standard guidance. This is probably based on the fact that even if there is no direct intended influence on a considered control process, there could be an influence due to a manipulation or malfunction in an adjacent system. This extends to the consideration of the system and the system architecture and the distribution and allocation of functions among the involved systems and elements.
Initiation of the safety/security engineering The initiation of the safety lifecycle is missing from the activities required by STPA-Sec. This step is used to clarify responsibilities and tailor the safety/security lifecycle for the development and is therefore in our opinion outside of the scope necessary for a analysis method.
Hazard and Threat Analysis and Risk Assessment Automotive safety and security engineering follows a risk based approach. The objective of the hazard and threat analysis and risk assessment is to identify and categorize hazards and threats, determine potential causes and formulate the safety and security goals for preventing or migrating them. The safety and security goals are based on the concept of avoiding unreasonable risks and reducing risks to a tolerable level. SAE J3061 requires additionally a identification of threats to Personally Identifiable Information (PII), this is currently not contained in STPA-Sec.
After the identification and classification of hazards and threats scenarios, STPA-Sec can be used to derive a set of constraints for a system to avoid such scenarios. Constraints, which restrict the system behavior, are only a part of the safety goals. Other concepts which ensure safe and secure operation like a transition to a safe state with reduced functionality or connectivity or fault/intrusion tolerant solutions are not directly mappable to constraints.
Functional safety/security concept The functional safety and security concept defines the functional safety and security requirements necessary to fulfill the safety and security goals. While a subset of the requirements can be derived from the constraints defined in STPA-Sec, there are additional aspects addressed in a functional safety and security concept. This is partially caused by different approaches towards safety and security engineering currently pursued in the standards and in STPA-Sec. The standards follow the strategy of fault or

vulnerability prevention and fault or vulnerability tolerance [20]. This includes more than trying to limit a system to only execute safe actions, the STPA approach. There are some overlaps between fault prevention and constraining a system to safe actions but this is currently a rather unexplored area.

6 Conclusion

In this paper we discussed the limitations of STPA-Sec and proposed several improvements for using it for top-down identification and analysis of unsafe and insecure scenarios for the concept phase in CPS design and development. We applied our proposed improvements in an automotive Battery Management System case study. Despite the improvements suggested in this paper, there are still open issues and it is very likely that a single approach will not be sufficient to satisfy all needs during the safety and security engineering of CPS. In general, STPA-Sec is suitable in the concept phase for safety & security co-engineering, but the results could be improved by combining the method with other approaches that focus more on the network and system architecture. In addition, STPA-Sec requires additional methods for the identification of potential hazards and the rating of risks. It is also still unclear if STPA-Sec is capable of handling large systems with a higher complexity and size.

There are some issues in STPA-Sec that do not align completely with current safety and security standards and activities. This requires extensions and the use of additional methods when developing a system in a standard-conform way. This is also addressed in ISO26262 with the requirement to use both, top-down and bottom-up analysis techniques. Addressing the aforementioned issues will be our research activities in the immediate future.

Acknowledgement This work is partially supported by EU ARTEMIS project AMASS (contract no. 692474) and Austrian Research Promotion Agency FFG on behalf of Austrian Federal Ministry of Transport, Innovation and Technology BMVIT.

References

1. William Young and Nancy Leveson, "Systems thinking for safety and security," in *Proceeding ACSAC '13*, pp. 1–8, ACM Press, 2013.
2. N. Leveson, "A New Accident Model for Engineering Safer Systems," *Safety Science*, vol. 42, pp. 237–270, Apr. 2004.
3. N. Leveson and J. Thomas, "An stpa primer," *Cambridge, MA*, 2013.
4. ISO, "ISO 26262–Road vehicles-Functional safety," 2011.
5. SAE, "J3061 Cybersecurity Guidebook for Cyber-Physical Vehicle Systems," 2016.
6. G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner, "Sahara: a security-aware hazard and risk analysis method," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*, pp. 621–624, IEEE, 2015.

7. C. Schmittner, T. Gruber, P. Puschner, and E. Schoitsch, "Security application of failure mode and effect analysis (FMEA)," in *Computer Safety, Reliability, and Security*, pp. 310–325, Springer, 2014.
8. C. Raspotnig, P. Karpati, and V. Katta, "A combined process for elicitation and analysis of safety and security requirements," in *Lecture Notes in Business Information Processing*, vol. 113, pp. 347–361, Springer, 2012.
9. M. Steiner, P. Liggesmeyer, and others, "Combination of safety and security analysis-finding security problems that threaten the safety of a system," in *Computer Safety, Reliability, and Security*, 2013.
10. M. Masera, I. Nai Fovion, and A. De Cian, "Integrating cyber attacks within fault trees," vol. 94, no. 9, pp. 1394 – 1402, 2009.
11. M. Bouissou and J.-L. Bon, "A new formalism that combines advantages of fault-trees and markov models: Boolean logic driven markov processes," *Reliability Engineering & System Safety*, vol. 82, no. 2, pp. 149–163, 2003.
12. ISO/IEC, "ISO/IEC 27000 Information technology - Security techniques - Information security management systems - Overview and vocabulary."
13. Charlie Miller and Chris Valasek, "Adventures in Automotive Networks and Control Units," (Las Vegas), 2013.
14. D. Kundur, X. Feng, S. Liu, T. Zourntos, and K. L. Butler-Purry, "Towards a framework for cyber attack impact analysis of the electric smart grid," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 244–249, IEEE, 2010.
15. S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular platooning in an adversarial environment," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pp. 167–178, ACM, 2015.
16. M. Krotofil, J. Larsen, and D. Gollmann, "The process matters: Ensuring data veracity in cyber-physical systems," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pp. 133–144, ACM, 2015.
17. J. Chynoweth, C.-Y. Chung, C. Qiu, P. Chu, and R. Gadh, "Smart electric vehicle charging infrastructure overview," in *Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, 2014.
18. Antuan Goodwin, "2011 Kia Optima Hybrid review: 2011 Kia Optima Hybrid." `http://www.cnet.com/products/2011-kia-optima-hybrid/`, June 2011.
19. Antuan Goodwin, "2015 Ford Focus Electric review: Ford keeps its electric car in Focus by lowering the price." `http://www.cnet.com/products/2015-ford-focus-electric/`, November 2014.
20. F. Ye and T. Kelly, "Component failure mitigation according to failure type," in *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, pp. 258–264, IEEE, 2004.