STPA-SafeSec: Safety and Security Analysis for Cyber-Physical Systems

Journal of Information Security and Applications 34 (2017) 183–196

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

STPA-SafeSec: Safety and security analysis for cyber-physical systems

Ivo Friedberg[a,b,*], Kieran McLaughlin[b], Paul Smith[a], David Laverty[b], Sakir Sezer[b]

[a] AIT Austrian Institute of Technology, Vienna, Austria
[b] Queen's University Belfast, Belfast, United Kingdom

ARTICLE INFO

Article history:
Available online 30 June 2016

Keywords:
Smart grid
Synchronous islanded generation
STPA
CPS
Safety
Cyber security

ABSTRACT

Cyber-physical systems tightly integrate physical processes and information and communication technologies. As today's critical infrastructures, e.g., the power grid or water distribution networks, are complex cyber-physical systems, ensuring their safety and security becomes of paramount importance. Traditional safety analysis methods, such as HAZOP, are ill-suited to assess these systems. Furthermore, cybersecurity vulnerabilities are often not considered critical, because their effects on the physical processes are not fully understood. In this work, we present STPA-SafeSec, a novel analysis methodology for both safety and security. Its results show the dependencies between cybersecurity vulnerabilities and system safety. Using this information, the most effective mitigation strategies to ensure safety and security of the system can be readily identified. We apply STPA-SafeSec to a use case in the power grid domain, and highlight its benefits.

1. Introduction

Critical infrastructures, e.g., the power grid or water distribution network, are cyber-physical systems (CPS): physical processes and components are connected over information and communication technologies (ICT), which are critical for correct system operation. As computing power and network transmission speeds increase, new applications for industrial control systems (ICS) build on advances in ICT to improve the efficiency of the underlying physical systems. These new applications create a tighter integration between physical processes and the cyber domain. Furthermore, systems are becoming more interconnected and thus more complex.

It is important to analyze the implications this increased use of ICT and resultant complexity has on the safety of these cyber-physical systems. This is necessary to ensure that safety requirements are identified and addressed as part of the system design process. Alongside safety aspects, cybersecurity threats to cyber-physical systems are becoming a concern. The Stuxnet virus (Karnouskos, 2011) or an attack on a German steel mill (Lee et al., 2014) showed how successful cyber-attacks can cause physical damage. Additionally, in the energy domain, research by Kang et al. (2015) has demonstrated how a multistage cyber-attack could result in the manipulation of a photovoltaic inverter, changing its active power output.

Traditional analysis methods that aim to assess the safety of critical infrastructures fail to encompass the complexity of emerging cyber-physical systems (Leveson, 2011). They work with accident models that are based on the *fault-error-failure* chain (see Avizienis et al., 2004). While these models are valid to describe failures of linear systems and single components, they are insufficient to describe system failures in complex interconnected systems. To tackle this shortcoming, Leveson (2004) has developed the Systems-Theoretic Accident Model and Processes (STAMP) accident model. Based on STAMP, the System Theoretic Process Analysis (STPA) approach was developed as a new hazard analysis technique to evaluate the safety of a system.

Current approaches to examining cybersecurity for cyber-physical systems are often based on an analysis of ICT protocols or network configurations, and are therefore strongly biased by information security concerns (Young and Leveson, 2014). Additionally, the effects of cyber-attacks need to be analyzed from a safety perspective. In this way, the potential impact of cyber-attacks on the safety of physical processes can be identified. To accurately quantify these impacts, an understanding of the relationship between cyber-attacks and physical processes is needed, requiring dedicated safety and security analysis techniques. In order to address this issue, Young and Leveson (2013, 2014) have developed STPA-sec, which uses the fundamental principles of STPA in the security domain. However, we will show in Sect. 3.1 that their approach —

[*] Corresponding author. Queen's University Belfast, Belfast, University Road; BT7 1NN, United Kingdom. Fax: +44 28 90 97 1702.
E-mail address: ifriedberg01@qub.ac.uk (I. Friedberg).

http://dx.doi.org/10.1016/j.jisa.2016.05.008

which indicates a separate analysis method for safety (STPA) and security (STPA-sec) — needs to be improved and clarified.

In this work, we present a novel analysis methodology that integrates safety analysis (STPA) and security analysis (STPA-sec) into one concise framework: STPA-SafeSec. Further, we overcome limitations of STPA by introducing security constraints to the analysis and by mapping the abstract control level of the system that is analyzed by STPA to real components. STPA-SafeSec provides a number of benefits over existing work: (i) it provides a single approach to identify safety and security constraints that then need to be ensured by the system in order to operate loss free. This single approach allows the interdependencies between safety and security constraints to be detected and used in mitigation strategies; (ii) the most critical system components can be prioritized for in-depth security analysis (e.g. penetration testing); (iii) the results from the analysis show the potential system losses that can be caused by a specific security or safety vulnerability in the system; and (iv) mitigation strategies can be more readily designed and their effectiveness evaluated — changes in the physical process can be used to mitigate cyber-attacks, while control algorithms can mitigate safety limitations of the physical processes or devices.

The rest of this work is structured as follows. In Section 2 we summarize previous work and highlight the shortcomings that our work overcomes. Section 3 motivates our approach before we present STPA-SafeSec in detail. To evaluate STPA-SafeSec we apply it to a use case in the power grid domain. This use case is described in Section 4, the results of the analysis in Section 5. Finally, Section 6 provides a discussion of the results before Section 7 concludes the paper.

2. Related work

Two groups of hazard analysis techniques can be identified: Failure-based hazard analysis techniques and systems-based hazard analysis techniques. Examples for failure-based techniques are Fault Tree Analysis (FTA) (Watson, 1961) and Failure Modes and Effect Analysis (FMEA) (Duckworth and Moore, 2010; Standard, 1980). Failure-based methods focus on the identification of the effects and probabilities of single component failures. The probabilities of component failures allow failure-based techniques to be quantifiable. However, the effectiveness of failure-based techniques and the increasing complexity of modern systems led to a new type of accidents rooted in the interaction of components. In order to understand this new type of accidents, system-based analysis techniques are needed. One such technique is the Hazard and Operability Analysis (HAZOP) (Lawley, 1974). Based on a firm system design, relevant system parameters are defined and guidewords are then used to identify how the system could deviate from the designed behavior. Over the years, researchers tried to formalize HAZOP to achieve objective and quantifiable results. But as Dunjó et al. (2010) highlight, all approaches to quantify the results led back to the use of failure-based FTA. System-based approaches are difficult to quantify because modern systems rely increasingly on software systems to control physical processes and are further embedded in socio-technical environments. Both software bugs as well as the effects of non-technical influences on the system over time are very hard to measure at design time. STPA aims to address these new challenges in a new system-based hazard analysis technique. Although there is no systematic approach to quantify the results of STPA, Thomas (2013) provides a mathematical model underlying STPA and a procedure to perform a STPA analysis systematically in his thesis. This systematic execution of STPA allows more objective analysis results compared to the adhoc application in recent years.

Research on the potential effects of cyber-attacks on physical infrastructures in CPS is often performed from either an ICT or a physical and control engineering perspective.

In the ICT domain, Dondossola et al. (2008) analyze potential cyber-attacks on power substation control systems. Their research focus lies on the potential threats of cyber-attacks to the ICT communication capabilities. Potential physical effects are highlighted but no critical physical effects are achieved in the experiments. Research like this is important to show that specific vulnerabilities that are well known from the cyber domain can be exploited in the context of a CPS. However, when physical processes get enriched by ICT based communication, the physical safety measures stay in place. These safety mechanisms are often used as an argument to downplay the risk that cyber-attacks can have on the safe operation of these systems. STPA-SafeSec considers existing safety mechanisms during the analysis process and can therefore make a stronger argument why cyber security vulnerabilities should be taken seriously.

Wang and Lu (2013) highlight cyber-attacks on availability, integrity and confidentiality, and their potential effects on different use cases and further present potential mitigation strategies. Signatures and encryption are presented as cryptographic countermeasures, and the difficulties that arise from limited computing power and strict time constraints are explained. Network-based countermeasures are presented and grouped by the targeted communication layers. The analysis provides a good overview on the necessary security considerations for CPS, but it is applied to a very generic smart grid architecture. This makes it hard to draw conclusions from the findings to a specific system at hand. STPA-SafeSec incorporates the generic aspects of these findings into an analysis framework that an engineer can apply to a specific system.

Kundur et al. (2011) present a first step toward a graph-based framework for modeling the physical impact of cyber-attacks on smart grids. Two case studies show the application of the framework in a Matlab environment based on two modified models of the IEEE 13 node distribution system. One case study shows that a successful cyber-attack can cause a severe under-frequency situation that ultimately results in a local blackout. For the power grid, the application of Petri-nets and attack trees to predict the load loss caused by identified weaknesses was presented by Ten et al. (2008). Laprie et al. (2007) use state machines to show how transitions caused by attacks can cause escalating, cascading and common cause failures. The use of mathematical system models has the big benefit that they can be validated and executed in an automated fashion. But their design is very time consuming, the modeling techniques are not standardized and often no tool support exists. For a system engineer who already has to perform a safety analysis of the system, this often means incalculable time effort that is not compulsory in most cases. STPA-SafeSec can be used instead of another safety analysis technique and includes the security analysis as a side product.

Srivastava et al. (2013) describe how an attacker can model cyber and physical vulnerability of a grid to cyber-attacks based on limited information. The authors present a cyber vulnerability ranking and how limited information about the grid can be used to identify the most critical components to launch an attack. In Ten et al. (2010) the authors present an automatic cyber-security resilience framework for power grids that incorporates detection, reasoning and automatic mitigation actions with a focus on the use of attack-trees for vulnerability analysis. In order to implement this resilience framework for a specific system architecture, a detailed analysis of the cyber-security vulnerabilities, their potential physical effects and the possible mitigation strategies is needed. The work by Ten et al. does not describe how this information should be acquired. STPA-SafeSec is designed to retrieve exactly this information from a system.

Sridhar et al. (2012) present a domain independent risk assessment methodology for cyber-physical infrastructures. The risk analysis starts with a detailed vulnerability assessment of the infrastructure followed by an analysis of application and physical system impact. Performing a complete vulnerability analysis of a complex CPS like a power grid is very complex and completeness is hard to prove. A better approach is to apply a detailed vulnerability analysis only at the most critical components identified by a previous impact analysis. That way, time consuming tasks like penetration testing can be scheduled most effectively. Further, penetration testing of a live CPS is ill-advised when the potential effects of invalid component behavior is not known. STPA-SafeSec provides the means to identify the most critical system components for in-depth security analysis. It further highlights the potential system hazards and system losses that can be caused by the malfunctioning of a specific component.

Availability attacks on GPS signals by GPS jamming are presented by Hu and Wei (2009). The authors also elaborate on countermeasures against GPS jamming in modern GPS receivers. Zhang et al. (2013) presents an integrity attack on GPS signals with respect to time synchronization. The authors show how the injection of targeted GPS signals increases the effect of the attack in comparison to the arbitrary error introduced by availability attacks. The relevance of such research on a specific CPS is often hard to identify. Whether there are devices deployed in the infrastructure that rely on GPS and what the impact of a jammed or manipulated signal would be are questions that are hard to answer. STPA-SafeSec provides the means to identify the dependencies of specific components on specific communication links and the impact of different attack categories.

Attack trees are one of the best established graphical security models. First introduced by Weiss (1991) as threat logic trees, their similarity to fault trees (Vesely et al., 1981) indicates that their roots are in the safety domain. Salter et al. (1998) first used the term attack tree, while Kordy et al. (2014) provide a timely summary of the different approaches that were developed since then. Given the dominance of tree structures in both, the safety and the security domain, a tree structure is leveraged by STPA-SafeSec to connect and present the final analysis results. The tree can then be extended by the results of an in-depth security analysis. Further, approaches to attach cost factors or probabilities to traditional attack trees have the potential to make the results of STPA-SafeSec quantifiable.

3. The STPA-SafeSec approach

3.1. Motivation

Cyber-physical systems (CPS) rely increasingly on the interconnectivity of devices. This caused increasing attention for cybersecurity alongside traditional analysis techniques. The presentation of STPA-sec by Young and Leveson (2013, 2014) was an attempt by the community around STAMP and STPA toward addressing this requirement. STPA-sec shows that STPA can also be used to analyze the security of systems. It changes the traditional bottom-up approach to security — where threats are used to derive the security requirements — to a top-down approach where the outcomes are relevant (Young and Leveson, 2013). A top-down approach could also be supported by other safety analysis techniques (e.g. FTA, HAZOP) but Sect. 2 highlighted the benefits of STPA over these approaches.

However, STPA can be used for more than just the analysis of system safety or security to prevent hazards and losses. It can be further generalized to analyze a system with respect to all relevant emerging system properties. An emerging system property is a property that arises through the interaction of parts of the system, while the parts themselves do not necessarily have the same property. Safety and security are examples of such emerging system properties. Previous publications on STPA-sec (Young and Leveson, 2013, 2014) suggest that security analysis is always performed to ensure system safety. Further, the publications indicate that there is a difference between the use of STPA for safety and STPA-sec for security. Sect. 1 argues that safety and security need to be addressed collectively. This collective approach is possible with STPA. However, the current presentations of STPA have a set of limiting factors that are addressed in this work.

(i) It is not made clear that STPA and STPA-sec are the same analysis. Only a collective analysis of safety and security allows the analyst to identify the dependencies between both properties and derive the most optimal results.
(ii) There is no guidance to an integrated approach of safety and security using STPA where safety and security are considered equally important properties that affect each other. Rather, STPA-sec argues that security is only relevant with respect to its impact on safety. This is a limited view on the system. In the socio-technical context of modern cyber-physical systems monetary loss to the operator should be considered a critical system loss. This loss can occur by a breach of confidentiality (e.g. consumer data or intellectual properties) without direct implications for safety. Subsequently, traditional STPA-sec needs to be extended to enable the analyst to consider this type of losses that are not directly safety related.
(iii) STPA-sec does not provide guidance on how to perform the security analysis once the critical system aspects are defined. First, it does not extend the causal factors presented for the safety domain into the security domain. This makes the analysis harder and limits comparability between different analysis results. Further, STPA-sec does not provide any means to integrate well established security analysis techniques. This is a problem not only for the acceptance of STPA for security analysis. It is also important for the quality of the analysis to get detailed results from penetration testing for example. Not all security constraints can be ensured in the physical and safety driven system domain. To derive the most effective system design STPA has to be able to guide manual analysis and integrate the results.

A general overview of STPA-SafeSec is given in Fig. 1. It aims to address the above mentioned shortcomings and introduces the following improvements over tradition STPA methods.

(i) The description and evaluation of a unified approach to safety and security analysis based on STPA and STPA-sec. This approach prioritizes safety and security equally and allows to detect a broader set of hazard scenarios.
(ii) An extension of the safety focused causal factor guidance of STPA into the security domain. This extension provides support for the analyst and makes the results more comparable.
(iii) A method to link the abstract control structure to the physical system design to integrate the results from traditional security analysis methods. Based on the physical system design, traditional security analysis techniques can be used complementary to STPA-SafeSec. Further, the results allow an intelligent application of time consuming analysis tasks to the most critical parts of the system.

The approach contains two loops. Modern cyber-physical systems are not static but evolve over time, influenced by their socio-technical environment. The outer loop highlights that STPA-SafeSec is an iterative approach that needs to be reapplied through the lifetime of the system to manage this evolving nature. In its core, STPA is based around constraints and the control loops of the system. To manage the complexity of modern systems, each control loop is

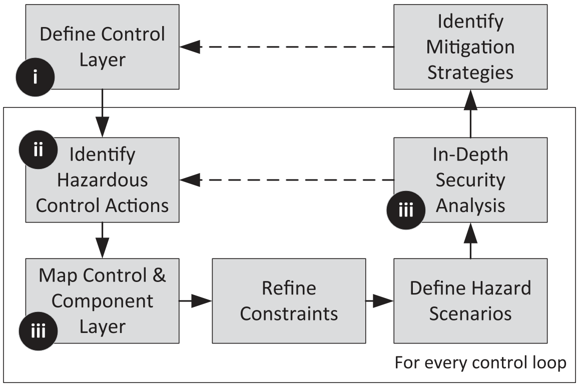

Define Control Layer
i
Identify Mitigation Strategies
ii
Identify Hazardous Control Actions
In-Depth Security Analysis
iii
Map Control & Component Layer
iii
Refine Constraints
Define Hazard Scenarios
For every control loop

Fig. 1. A simplified view on the steps that are performed during an STPA-SafeSec analysis. The solid arrows indicate the order in which the steps are taken. Dashed arrows indicate where STPA-SafeSec is iterated. Black circles label at what steps the extensions of STPA-SafeSec are applied. At first, the control layer is defined for the whole system that is comprised of different control loops. Each control loop is then analyzed in detail to define scenarios of how system flaws or malicious actions can cause system hazards. Based on all scenarios, the most effective mitigation strategies in the system can be identified.

analyzed separately during the analysis; this is shown by the inner loop.

The constraints are first refined for the system as a whole based on the losses that should be prevented. Then, these constraints get refined and mapped to the control layer; the representation of the control loops and their interaction. Hazards and subsequent losses happen, when control actions that violate one or more of the previously defined constraints — so called hazardous control actions — are taken. The analysis provides the means to derive the causal factors that lead to these hazardous control actions. These causal factors are extended by STPA-SafeSec to encompass security considerations. Further, the abstract control layer gets mapped to an implementation specific component layer. This component layer provides the means to further refine the constraints and derive more specific causal factors. It can further guide in-depth security analysis. The final results — a set of changes that need to be applied to the system architecture to ensure loss free operation — are finally derived based on the scenarios that describe how hazardous control actions are caused due to causal factors.

3.2. STPA-SafeSec approach in detail

Beside the description of a unified approach to the analysis, STPA-SafeSec provide two core contributions to the analysis. The first contribution is the description of a *generic component layer diagram* shown in Fig. 2. To evaluate whether security constraints are ensured, the focus of the analysis needs to expand from the more abstract control layer of the system to a new component layer. STPA introduced the *generic control structure diagram* to help the analyst to identify the control loops of a system. While the control layer focuses on functional interactions, control concepts and algorithms, the component layer visualizes the system implementation. This includes the nodes at which control algorithms or sensors are deployed as well as network nodes, physical network connections and application level protocols used.

The second contribution is the extension of causal factors into the security domain. STPA provides a *generic causal factor diagram* that presents the factors that lead to hazardous control actions. However, the factors do not include active actions with malicious intent. We derived a set of factors that describe the effects of cyber attacks on integrity and availability shown in Tables 1 and 2. In a first step, this extension provides the analyst a list of factors to consider. However, in combination with the component layer, STPA-SafeSec also provides a guidance for the mapping of causal factors between the control and the component layer. For each attack effect, our guidance shows the capabilities that are required by the attacker at component layer to achieve the malicious effect. For example, if missing feedback is identified as a causal factor for a hazardous control action, availability constraints should be placed at the physical communication network between sensor and controller as well as on the sensor's feedback mechanism. Table 2 can then show that either network nodes, end-nodes or the physical communication link can be attacked to achieve a delay or loss in communication.

STPA-SafeSec relies on these two extensions to provide a unified analysis approach shown in detail in Fig. 3. The two loops from Fig. 1 can be seen again. The iterative application of the top-down analysis to cover the effects of the socio-technical environment on the system is shown by the *System Refinement* loop. Within this analysis, the *Control Loop Analysis* loop is used to manage the system complexity. The control loops can be considered in a hierarchical structure where the simple loops that have no further dependencies build the lowest level.

Each step during the analysis produces *artifacts*. These artifacts are the result of the analysis and allow to identify shortcomings in the system design and to trace them back to system losses. The analyst first identifies high level system losses, before the system hazards are identified based on these losses. In traditional STPA, safety constraints are then derived from the set of identified hazards (see also steps *II-IV* in Fig. 3). In STPA-SafeSec, system-wide security constraints are defined alongside safety constraints. Once the constraints are identified, a *control layer* is built in step *V*. This control layer is the counterpart to the *safety control structure* in STPA. It is a graphical representation of the control loops in the system and the interactions between the different controllers. Further, the safety and security constraints are considered collectively and mapped to the different aspects of this control layer. The control layer builds the basis for further steps of the analysis.

The *Control Loop Analysis* loop aims to identify causal factors in the system operation that lead to a violation of either safety or security constraints. As shown by Fig. 1, hazardous scenarios are defined that describe in detail how a specific constraint gets violated. Steps *IV-XIII* describe in detail how each control loop is analyzed in order to identify these hazardous scenarios. Based on the hazardous scenarios, mitigation strategies can be applied to the system design to address the initially identified safety constraints (*XV*).

Hazardous control actions can be defined in a semi-automated manner as defined by Thomas (2013). Based on the set of hazardous control actions STPA-SafeSec leverages the *generic causal factor diagram* as well as the set of integrity and availability threats presented in Tables 1 and 2 to identify the causal factors. Step *XI* in the flow diagram (Fig. 3) is introduced with STPA-SafeSec to generate a mapping between the control level representation of each control loop and its component level representation. This mapping shows what algorithms run on which physical components, and which networks are used for each abstract communication path. In steps *XI* and *XII* — before the hazardous scenarios are identified — both safety and security constraints are refined and mapped to the component layer. This allows the analyst to enrich safety constraints on the control layer with security constraints at component layer; again Tables 1 and 2 can be used for guidance.

Based on this information, hazardous scenarios are derived. Because constraints are derived from each other, hazardous scenarios can be refined and represented in a hierarchical tree structure. In the example of feedback availability, the root scenario would be a violation of the safety constraint that states that the feedback has to be available. Child scenarios in the security do-

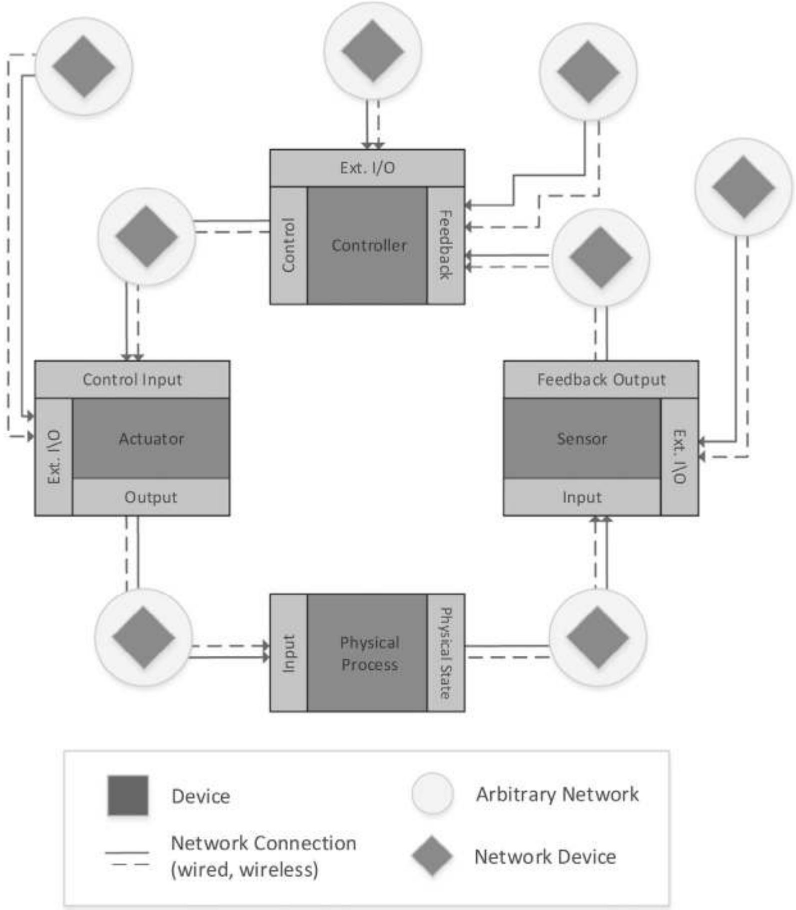


Fig. 2. Generic component layer diagram for a simple control loop. This diagram shows the potential physical components involved in a control loop and can be used together with Tables 1 and 2 to identify security constraints and perform in-depth security analysis.

Table 1
General integrity threats. A full circle indicates that attack capabilities on the specified entity are sufficient to exploit the threat. Half circles indicate capabilities needed but not sufficient.

#	Description	■	◆	═	Protocol
CSTR-I-1	Command injection	●	◐	◐	◐
CSTR-I-2	Command drop	●	●	●	○
CSTR-I-3	Command manipulation	●	◐	○	◐
CSTR-I-4	Command delay	●	●	●	○
CSTR-I-5	Measurement injection	●	◐	◐	◐
CSTR-I-6	Measurement drop	●	●	●	○
CSTR-I-7	Measurement manipulation	●	◐	○	◐
CSTR-I-8	Measurement delay	●	●	●	○

Table 2
General availability threats.

#	Description	■	◆	═	Protocol
CSTR-A-1	Communication delay	●	●	●	○
CSTR-A-2	Communication dropped	●	●	●	○
CSTR-A-3	Node overloaded (delay)	●	●	○	○
CSTR-A-4	Node overloaded (drop)	●	●	○	○

main represent a violation of the availability constraint on each relevant component respectively. Further scenarios can be identified in the safety domain with respect to reliability of sensors for example.

Finally, the component layer of the system can be used in step *XIV* to guide a detailed security analysis. Based on the most critical constraints at the control level, the constraints at the component level determine which components should be prioritized for in-depth security analysis. Identified vulnerabilities can be seen as potential violations of these security constraints. Thus, the potential system losses that can be caused by various cyber-attacks can be traced through the scenario tree to potential system losses. It is further possible to identify the most effective mitigation strategies. A set of mitigation strategies is effective if it mitigates all paths in the scenario trees to a root scenario. The correlation between safety and security constraints, as well as the mapping between control and component layer, are enablers for mitigation strategies and to identify the minimal set that is most effective. For example, security related vulnerabilities might not be resolvable when legacy equipment cannot be replaced. Depending on the vulnerability, a mitigation strategy at the control layer can instead mitigate the violation of the related safety constraint. In this situation, an attack would still be possible, but a hazard cannot be achieved by the attacker.

STPA-SafeSec addresses the shortcomings of STPA and STPA-sec that were presented in Sect. 3.1. It provides one integrated approach to analyze safety and security aspects in a single framework. That way it eliminates the need to perform steps of the analysis repeatedly and reduce the risk of misunderstandings. STPA-SafeSec guidance for the analyst during the identification of security constraints and allows a better prioritization of time consuming tasks during the detailed security analysis phase (e.g. penetration testing). Finally, the integration of safety and security constraints into one multi-layered representation of the system highlights the dependencies between the two domains.

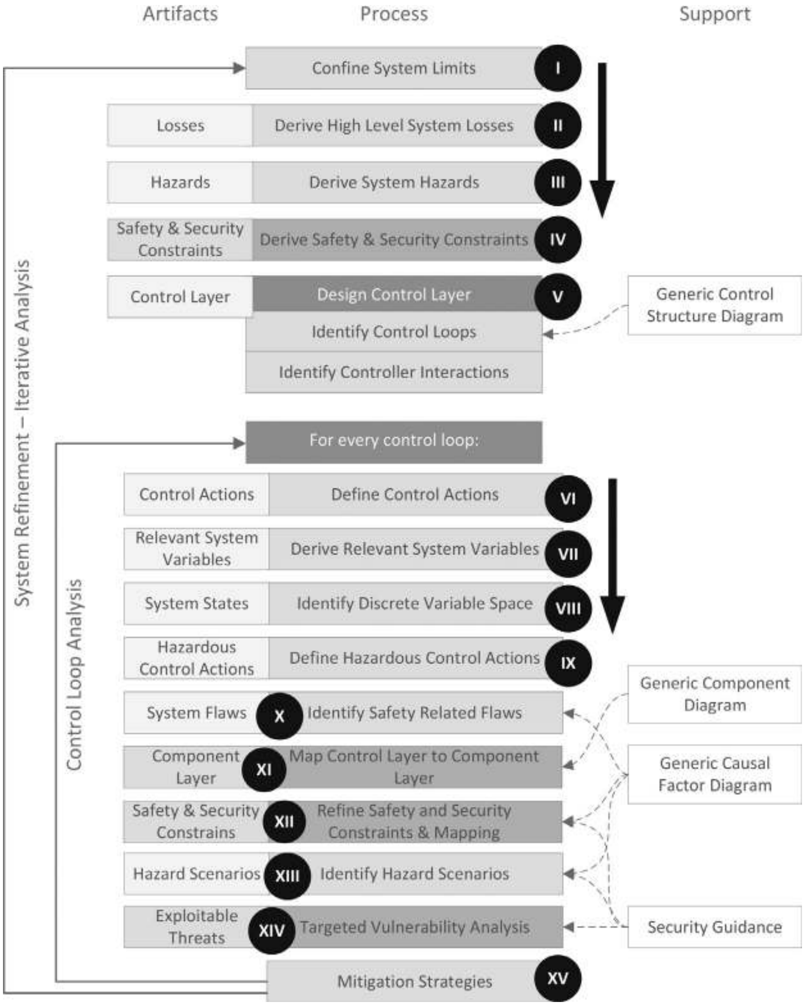

Artifacts
Process
Support
Confine System Limits
I
Losses
Derive High Level System Losses
II
Hazards
Derive System Hazards
III
Safety & Security Constraints
Derive Safety & Security Constraints
IV
Control Layer
Design Control Layer
V
Generic Control Structure Diagram
Identify Control Loops
Identify Controller Interactions
System Refinement – Iterative Analysis
For every control loop:
Control Loop Analysis
Control Actions
Define Control Actions
VI
Relevant System Variables
Derive Relevant System Variables
VII
System States
Identify Discrete Variable Space
VIII
Hazardous Control Actions
Define Hazardous Control Actions
IX
System Flaws
X
Identify Safety Related Flaws
Generic Component Diagram
Component Layer
XI
Map Control Layer to Component Layer
Generic Causal Factor Diagram
Safety & Security Constrains
XII
Refine Safety and Security Constraints & Mapping
Hazard Scenarios
XIII
Identify Hazard Scenarios
Exploitable Threats
XIV
Targeted Vulnerability Analysis
Security Guidance
Mitigation Strategies
XV

Fig. 3. Flow diagram of integrated safety and security analysis based on STPA. The process in the middle describes the sequence of the tasks to be performed during the analysis. Each task can produce artifacts (to the left of a task) and can leverage support mechanisms from the framework (to the right of the task). Two loops highlight the iterative nature of the analysis.

4. Synchronous islanding use case

This section presents an introduction to the use case for the analysis performed in Sect. 5. It is based around the concept of synchronous-islanded operation microgrids. With the increasing penetration of the traditional power grid with renewable energy sources — e.g. photovoltaics (PV) or wind turbines — emerged the need for finer control capabilities. Farhangi (2010) and Considine et al. (2012) among others identify microgrids as one way to manage the complexity of the future smart grid. It allows for an effective integration of renewable energy sources, but introduces new requirements for the tighter integration of ICT resources. A microgrid is a potentially independent subset of any host grid where generation, storage and load are in close local proximity. It is characterized by its ability to operate either connected to the host grid or — for example in case of a fault on the host grid — as an independent power island. For most types of microgrids, independent operation is only possible for a limited amount of time. To prevent blackouts in the microgrid during reconnection, it has to be possible for microgrids to be dynamically added and removed from the host grid during operation. Synchronous-islanded operation of microgrids is seen as one way to tackle this challenge. Even in islanding mode, the power metrics — *voltage magnitudes* (X_m), *frequency* (ω) and *phase angle* (ϕ) — are kept synchronized with the host grid. When these power metrics are matched between the islanded microgrid and the host, circuit breaker re-closure is safe. If synchronization is not guaranteed, re-closure of circuit breakers has to be prohibited.

A universally applicable method for synchronous-islanding is presented by Best et al. (2008). The authors describe general requirements and possible limitations caused by time-delay introduced when transmitting the reference signal. Control logic is used to control the difference in frequency between the systems while it minimizes the current phase angle difference. Challenges like islanding detection and control initiation are covered as well as issues with power quality and security mechanisms in the case of a communication loss.

Synchronous-islanded operation enforces strict transmission delay constraints on the underlying communication network. Laverty et al. (2008) perform a detailed analysis on the effect of the time delay introduced by wide-area telecommunications. In their work, the authors show the response of an alternator operated by an Internet-based phase difference controller to local load acceptances. They were able to show that control is effective when it is operated on a telecommunication link with variable time delay such as the Internet. Caldon et al. (2004) evaluate the effects of synchronous and inverter-interfaced generators on the stability of power islands. The authors show that inverter-interfaced generators increase the stability of frequency and phase angle difference.

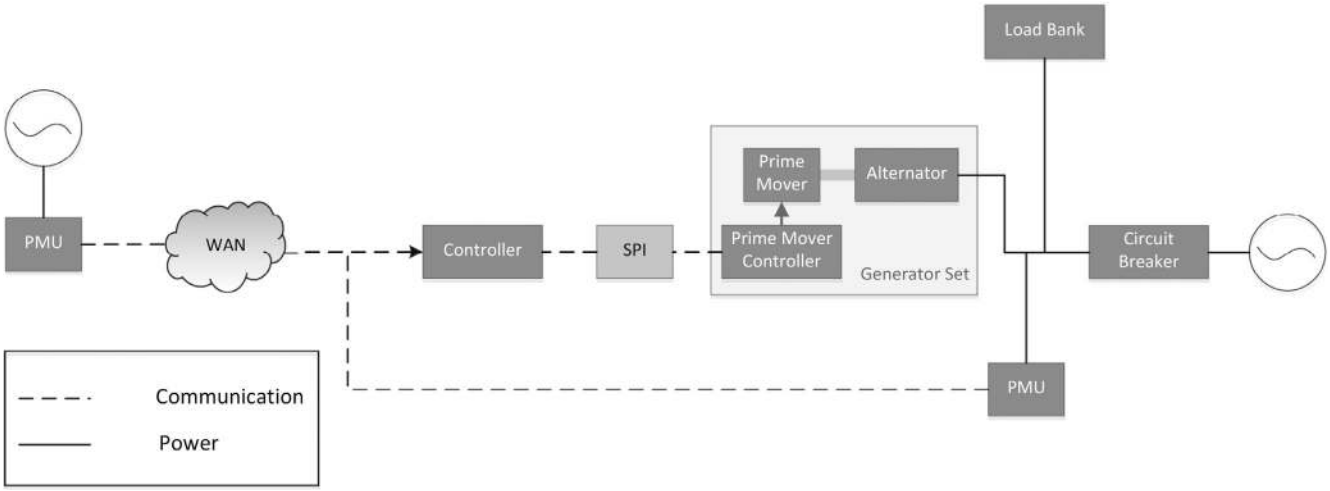


Fig. 4. Overview on the testbed architecture. A generator set is operated synchronous with the main grid while in islanding mode. A Phasor Measurement Unit (PMU) at a remote, secure location in the main grid communicates with a local controller. A second PMU measures the power metrics in the island. The controller compares the measurements from the two PMUs and controls the generator.

Fig. 4 shows a high level representation of the synchronous-islanding testbed that will be analyzed in this paper. It is designed to operate an alternator, driven by the prime mover (in this case a DC machine), in a synchronized manner with the main grid while in islanded mode. The **Generator Set** is a DC Motor/Alternator set. The DC motor is supplied from a 'Eurotherm 590+' digital prime mover controller; it offers analogue inputs to control the set points on the drive. **Phasor Measurement Units (PMUs)** are a measurement device that captures *voltage magnitudes* (X_m), *frequency* (ω) and *phase angle* (ϕ) periodically and transmits it to the configured receiver. For PMU measurements to be useful, they need to be taken at the exactly same time at every place in the grid. This clock synchronization is commonly achieved with GPS signals, although alternative methods are possible. In our testbed PMUs are deployed at the host grid and within the power island. The collected power metrics are then transmitted to the controller. The **Load Bank** is a 3-phase resistive load bank deployed within the power island. It can be used to evaluate the behavior of the controlled island under shifting loads. The **Controller** collects the measurements from the PMUs and adapts the set points of the generator set.

For a more detailed instruction on the use case we refer the reader to Friedberg et al. (2015).

5. Analysis

In this section we will apply STPA-SafeSec to the synchronous-islanding use case. The testbed model in Fig. 4 shows a high level view of the architecture. During our analysis we followed the detailed STPA-SafeSec flow diagram given in Fig. 3. This section is divided into subsections where each represents one step in the simple STPA-SafeSec diagram in Fig. 1. How hazard scenarios are used to guide the in-depth security analysis and how the results can be used to derive mitigation strategies will be covered in Sect. 6. During the analysis, we will use a set of acronyms to refer to the different artifacts that result form the analysis. Table 3 highlights the different acronyms and their meaning.

5.1. Defining the safety control structure

We define the system under analysis as the control of the generator set in the different system states. In the first steps we identify system losses and system hazards based on expert knowledge. There is no formal process provided but system losses and hazards are in general high level properties. We can define losses for our system as follows: **(L-1)** Injury to humans, **(L-2)** damage to power equipment, **(L-3)** damage to end-user equipment and **(L-4)** interruption of power supply to consumer loads. Based on the system losses, system hazards are identified: **(H-1)** Out-of-sync reclosure, **(H-2)** operation of power equipment outside of operational limits, **(H-3)** violation of power quality metrics, **(H-4)** inability to achieve synchronization and **(H-5)** inability to meet local demand. Hazard *H-2* can be further refined for the concrete system as: The generator set should not be operated at an analog set-point outside the range 0 V−5 V, since there is only one critical power equipment in the architecture; the generator set. Hazard *H-3* should be split into **H-3.1** (Power quality violation regarding voltage) and **H-3.2** (Power quality violation regarding frequency). Table 4 shows which losses are potentially caused by each hazard.

The next step is to derive high level safety and security constraints for the system under evaluation (see step *IV* in Fig. 3). The easiest way to identify high level constraints is by negation of the identified system hazards (*CSTR-S-1 – CSTR-S-5* respectively). The

Table 3
List of acronyms used to reference the different analysis artifacts throughout this work. *X* acts as a placeholder for the numbering.

Acronym	Description
L-X	System losses
H-X	System hazards
F-X	System flaws
HC-X	Hazardous control actions
CTRL-N-X	Node at control layer
CTRL-C-X	Connection at control layer
CPT-N-X	Node at component layer
CPT-C-X	Connection at component layer
CSTR-S-X	Safety constraint
CSTR-A-X	Availability constraint
CSTR-I-X	Integrity constraint

Table 4
Losses that can be caused by each identified hazard.

Hazard	L1	L2	L3	L4
H-1	x	x	x	x
H-2	x	x	x	x
H-3.1	x		x	
H-3.2			x	x
H-4				x
H-5				x

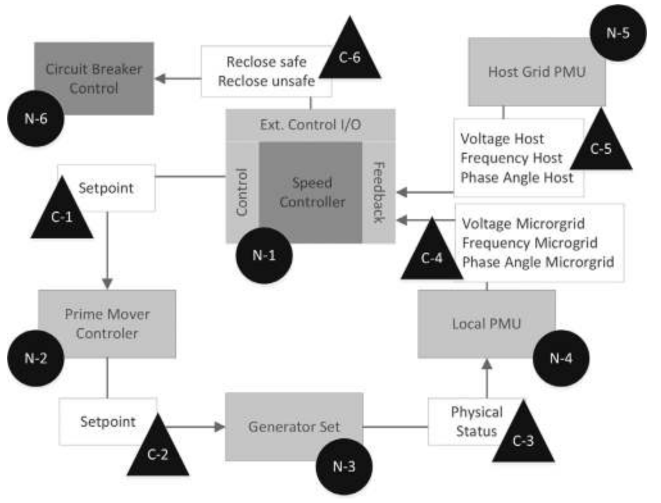


Fig. 5. Control layer diagram for synchronous-islanding test case. Gray blocks highlight logical nodes, while white blocks describe the type of variables and commands transmitted between the nodes. Black circles are labels to logical nodes. Block triangles are labels to logical connections. Each label has CTRL- as prefix in the text which is omitted in the figure for simplicity.

resulting constraints can be safety and security related, depending on the hazard. In the use-case at hand, no high-level security constraints were derived. However, this does not mean that we are considering a pure safety use case. At this first stage we look at the overall system operation and decide what we have to ensure to prevent losses from a top-down perspective. During the analysis we will refine these constraints and show that security needs to be considered as well to grasp the complexity of the system.

Fig. 5 gives a detailed view on the control level representation of the system under evaluation (step *V*). The traditional STPA analysis provides a generic control layer diagram to guide the generation of this control level representation. Fig. 5 is designed based on the control loop in Fig. 4. The equivalence between the two control loops is not obvious. In the original STPA documentation the flow of control commands and information in a control loop is represented in a counter clockwise fashion. Traditionally control loops are represented as a clockwise loop (as done in Fig. 4). STPA-SafeSec uses the STPA way of counter clockwise representation.

We identified one control loop — the *Speed Controller* — and its interaction with the different logical components in the system. Each node (CTRL-N) and each connection (CTRL-C) at the control layer are labeled to allow clear reference in the text. For simplicity, the prefix *CTRL-* is omitted in the figure. In the heart of the control loop is the *Speed Controller* (*CTRL-N-1*). It receives feedback from the two PMUs in the local microgrid (*CTRL-N-4*) and the host grid (*CTRL-N-5*) via two separate connections (*CTRL-C-4* and *CTRL-C-5*). Each PMU reports the measured voltage magnitude (X_m), frequency (ω) and phase angle (ϕ) periodically. Further, the controller checks if synchronization is achieved and circuit breaker reclosure is therefore safe or unsafe and transmits this information to the circuit breaker controller (*CTRL-C-6, CTRL-N-6*). The circuit breaker controller is not implemented as an automatic control loop due to safety reasons in the testbed. Instead a human operator acts as the circuit breaker controller. Finally, the speed controller can define a set point (*CTRL-C-1*) for the generator set (*CTRL-N-3*) over the DC Drive (*CTRL-N-2*).

5.2. Hazardous control actions

Based on the control layer diagram, Table 5 shows the identified system variables that are relevant for the correct operation of the speed controller (see steps *VII* and *VIII*). It further shows the control commands that the controller can issue. For each variable, the value space needs to be broken up into discrete steps that are safety relevant. This allows an automatic enumeration through the different system states to identify the potential hazardous scenarios later on (see also Thomas, 2013 for details). To achieve synchronization, the speed controller needs to calculate the difference in the power metrics between host grid and microgrid. We define the following variables as the difference of voltage magnitude, frequency and phase angle between the local microgrid and the host grid:

Table 5
System variables that are relevant for the correct operation of the speed controller and their possible values.

#	Name	Values
	Voltage magnitude difference	Within limits; outside limits
$\Delta_\omega(t)$	Frequency difference	Within limits; outside limits
$\Delta_\phi(t)$	Phase angle difference	Within limits; outside limits
	Command to prime mover	Within limits 0-5; outside limits
	Command to circuit breaker	CB reclosure safe; CB reclosure unsafe
	Circuit breaker status	Open; closed

$$\Delta_{X_m}\ (t) = \left|X_m^h\ (t) - X_m^\mu\ (t)\right| \tag{1}$$

$$\Delta_\omega\ (t) = |\omega_h\ (t) - \omega_\mu\ (t)| \tag{2}$$

$$\Delta_\phi\ (t) = |\phi_h\ (t) - \phi_\mu\ (t)| \tag{3}$$

For the difference in power metrics, two system states can be identified. The difference can either be small enough that circuit-breaker reclosure is safe (*within limits*) or not. For each possible control action a decision has to be made whether the control action is hazardous in a given system state or not. A control action in STPA can be hazardous if (i) it is applied at all, (ii) it is applied too early, (iii) it is applied too late or (iv) if it is not applied in a given system state. Table 6 shows a list of the hazardous control actions identified for the speed controller. A hazardous control action is present for a specific system state and can cause a set of hazards. *HC-1 - HC-3* describe the cases where the speed controller wrongfully assumes that synchronization is achieved. It would then indicate that the reclosure of the circuit breaker is safe while it is not. This can cause out-of-sync reclosure (*H-1*) or power quality violations (*H-3*). In this case, it does not matter when the control command to the circuit breaker controller (*CTRL-N-6*) is sent, as any occurrence of the command in this system state is potentially hazardous. *HC-4* highlights that the set-point to the generator set must not be outside the operational limits for the generator set. Again, the timing of the command is irrelevant, as is the state of the circuit breaker and the power quality metrics. Finally, *HC-5* shows that it is hazardous for the speed controller to miss a valid update of the set-point to the generator set if the circuit breaker is open. In this case, the local supply of local loads depends solely on the power generated by the generator set. Failure to adjust the power output to changing load situations can cause power loss, power quality violations and consequently damage to consumer equipment.

5.3. Map control to component layer

Now that we have identified the cases in which certain control actions cause system hazards, the goal is to identify the system flaws that enable hazardous control actions. In traditional STPA this is done solely in the safety domain. In our integrated approach, we also include the violation of security constraints as potential causal

Table 6
Hazardous control actions by different system states. A '–' indicates that the status of the variable is irrelevant for the hazardous behavior of the highlighted control action. A control action can either be hazardous at any time it is performed in a given state, or only if provided too early or too late or not at all.

#					Δ_ω	Δ_ϕ	Any time	Too early	Too late	Not	Hazards
HC-1	Safe	–	Open	Out	–	–	1	1	1	0	H-1, H-3
HC-2	Safe	–	Open	–	Out	–	1	1	1	0	H-1, H-3
HC-3	Safe	–	Open	–	–	Out	1	1	1	0	H-1, H-3
HC-4	–	Out	–	–	–	–	1	1	1	0	H-2
HC-5	–	In	Open	–	–	–	0	0	1	1	H-3, H-4, H-5

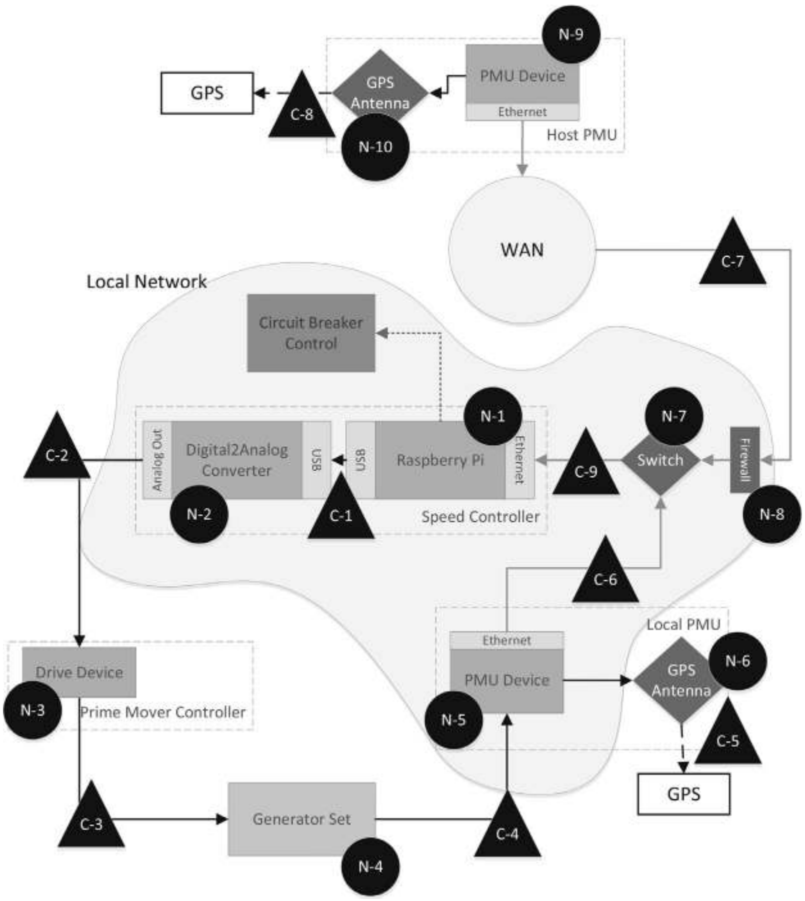
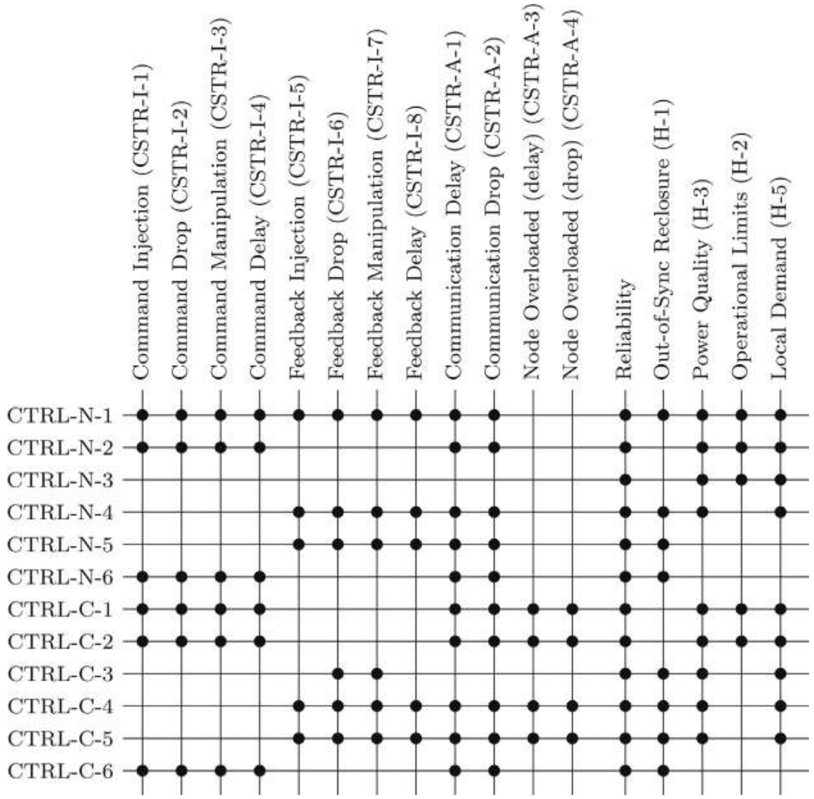


Fig. 6. Component layer diagram of synchronous-islanding use-case. Dashed boxes indicate the representation of components at control level. Solid lines are wired connections while dashed lines indicate wireless communication. Black lines show direct end-to-end connections. Gray lines show IP based transportation. Black circles label concrete nodes while black triangles label concrete connections.

factors. Therefore, we map the control layer given by Fig. 5 to the physical component layer. Fig. 6 shows the component layer for our synchronous islanding testbed. The nodes in Fig. 5 are still visible but represented by nodes at the component layer (CPT-N). The same is true for control layer connections that are now represented by connections at the component layer (CPT-C). Component layer nodes and connections are the architectural implementation of the high level control layer in the physical world. While the generator set (*CTRL-N-3*) is implemented by one physical component (*CPT-N-4*), the implementation of other nodes and connections is significantly more complex. The speed controller (*CTRL-N-1*) is actually implemented by two nodes. First, a Raspberry Pi (*CPT-N-1*) is used to handle the control algorithm and to manage the IP based communication to the local network (*CPT-C-7*). An additional microcontroller (*CPT-N-2*) is connected via serial link (*CPT-C-1*) and converts the digital control signal of the Pi to an analog signal required by the DC drive. This shows how one node at the control layer is represented by a combination of nodes and connections in the component layer. The representation of PMU devices (*CTRL-N-4, CTRL-N-5*) is also more complex. The component layer highlights the dependency of PMUs on a GPS signal. An additional GPS antenna component is introduced (*CPT-N-6, CPT-N-10*) together with a wireless communication link to the GPS system. Finally, the IP based communication between PMUs and speed controller is shown in more detail. Where the control level only highlighted the fact that a communication exists between PMUs and controller (*CTRL-C-4, CTRL-C-5*) the component layer highlights the full complexity. Network nodes like switches (*CPT-N-7*) and firewalls (*CPT-N-8*) are shown as well as different network enclaves. The component diagram also highlights that safety and security constraints can be placed in the local network where network components are known. The Wide Area Network (WAN) that is used to transport data from the host grid PMU on the other hand is not under control of the system architects. Therefore, constraints cannot be ensured and mitigation strategies need to be in place to ensure safety and security.

With the mapping between control and component level achieved we can identify high level design flaws that can lead



	Command Injection (CSTR-I-1)	Command Drop (CSTR-I-2)	Command Manipulation (CSTR-I-3)	Command Delay (CSTR-I-4)	Feedback Injection (CSTR-I-5)	Feedback Drop (CSTR-I-6)	Feedback Manipulation (CSTR-I-7)	Feedback Delay (CSTR-I-8)	Communication Delay (CSTR-A-1)	Communication Drop (CSTR-A-2)	Node Overloaded (delay) (CSTR-A-3)	Node Overloaded (drop) (CSTR-A-4)	Reliability	Out-of-Sync Reclosure (H-1)	Power Quality (H-3)	Operational Limits (H-2)	Local Demand (H-5)
CTRL-N-1	●	●	●	●	●	●	●	●	●	●			●	●	●	●	●
CTRL-N-2	●	●	●	●					●	●			●		●	●	●
CTRL-N-3													●		●	●	●
CTRL-N-4					●	●	●	●	●	●			●	●	●		●
CTRL-N-5					●	●	●	●	●	●			●	●			
CTRL-N-6	●	●	●	●					●	●			●	●			
CTRL-C-1	●	●	●	●					●	●	●	●	●		●	●	●
CTRL-C-2	●	●	●	●					●	●	●	●	●		●	●	●
CTRL-C-3						●	●						●	●	●		●
CTRL-C-4					●	●	●	●	●	●	●	●	●	●	●		●
CTRL-C-5					●	●	●	●	●	●	●	●	●	●	●		●
CTRL-C-6	●	●	●	●					●	●			●	●			

Fig. 7. Mapping between safety and security constraints and the logical nodes and connections at the control layer (see Fig. 5).

to hazardous control actions. Again we can leverage existing tools from traditional STPA. It provides a generic causal factor diagram to provide the potential design flaws that can have a safety implication (see Leveson, 2011, p. 223). After applying this to our use case we derived the following flaws in our system: **F-1:** Controller incorrectly believes that voltage difference is within limits; **F-2:** Controller incorrectly believes that frequency difference is within limits; **F-3:** Controller incorrectly believes that phase angle difference is within limits; **F-4:** Setpoints outside of the operational limits is received by DC Drive; **F-5:** Speed controller incorrectly believes that no setpoint correction is required; **F-6:** Circuit breaker controller incorrectly receives information that reclosure is safe. In traditional STPA the next step would be to identify hazardous scenarios (see step *XIII* in Fig. 3). Instead we first refine the safety and security constraints of the system based on the control and the component layer. To mitigate *F-1* - *F-3* as well as *F-5* and *F-6*, safety and security constraints need to be ensured on the feedback mechanisms of the system. For *F-6* further constraints are required at the controller as well as at the communication to the circuit breaker controller. *F-4* needs to be ensured at the controller and actuator as well as at the communication to the actuator and to the generator set.

5.4. Refine safety and security constraints

Based on the set of system flaws, Fig. 7 highlights what security and safety constraints can be violated at each node and connection at control level. High level security threats align with the threats presented in Tables 1 and 2 from Sect. 3.2. The set of safety constraints is taken from the hazards that can be caused at each node or connection.

To achieve a better understanding of how hazardous scenarios manifest, the constraints need to be mapped to the component layer. These steps are unique to STPA-SafeSec and build the core contribution to the analysis. They allow STPA-SafeSec to overcome the shortcomings of previous approaches. We will see that this is a fairly complex task that results in a lot of data. For space limitations, we will describe the constraint mapping only for the three most interesting cases.

5.4.1. The local PMU device

The first case is the local PMU device (*CTRL-N-4*). We can see from Fig. 7 that malfunction of *CTRL-N-4* can lead to out-of-sync reclosure, violation of power quality and failure to meet local demand. All three hazards can only be caused when the circuit breaker is open (see Table 6). The PMU is further expected to be reliable; under expected circumstances, the measurement quality is supposed to match the expectations on the device. A cyber-attack on *CTRL-N-4* can cause feedback to be unavailable to the controller (*CSTR-A-1* and *CSTR-A-2*) or violate the integrity of the feedback (*CSTR-I-5* - *CSTR-I-8*).

CTRL-N-4 is actually represented by two nodes at component layer. First there is the PMU device (*CPT-N-5*) but there is also a GPS antenna attached (*CPT-N-6*). The mapping of cyber-security constraints from control to component layer can be automated with the use of Tables 1 and 2. They show the implementation aspects that need to ensure a specific security constraint at the control layer. The STPA-SafeSec process derives that all security constraints on *CTRL-N-4* need to be ensured by *CPT-N-5*. Further analysis needs to be performed to ensure the safety constraints given the concrete implementation. The connection between *CPT-N-5* and *CPT-N-6* is hardwired and can be considered stable and secure as long as both endpoints are but we need to extend constraints to the GPS connection. Known attacks exist that can spoof (see Zhang et al., 2013) or jam (see Hu and Wei, 2009) a GPS signal. Therefore, *CSTR-A-1* and *CSTR-A-2* need to be ensured on *CPT-N-6* and *CPT-C-5*. Further, depending on the PMU implementation, failure to receive a correct GPS signal can cause invalid or miss-

Table 7
This table highlights which hazards can be caused by a violation of the different constraints at each concrete component and concrete connection mapped to by *CTRL-N-4*.

	CPT-N-5	*CPT-N-6*	*CPT-C-5*
CSTR-I-5	*H-1, H-3, H-5*	*H-3, H-5*	*H-3, H-5*
CSTR-I-6	*H-3, H-5*	*H-3, H-5*	*H-3, H-5*
CSTR-I-7	*H-1, H-3, H-5*	*H-3, H-5*	*H-3, H-5*
CSTR-I-8	*H-3, H-5*	*H-3, H-5*	*H-3, H-5*
CSTR-A-1	*H-3, H-5*	*H-3, H-5*	*H-3, H-5*
CSTR-A-2	*H-3, H-5*	*H-3, H-5*	*H-3, H-5*
Reliability	*H-1, H-3, H-5*	*H-3, H-5*	*H-3, H-5*

ing feedback. The controller will be able to detect an error in clock synchronization; therefore, out-of-sync reclosure can be prevented. But the lack of valid feedback will not allow the controller to stabilize the system when it is disconnected from the host grid. Violation of power quality and inability to meet local demand are possible. Therefore, the reliability of the PMU depends on the reliability of the GPS connection (*CPT-C-5*).

Table 7 shows a summary of the hazards that can be caused by a violation of each constraint at each concrete node and concrete connection. We can see that reliable time synchronization as well as timely availability of the local PMU measurements are necessary to achieve system stability in islanded operation. However, unavailable feedback, delays and missing clock synchronization can be detected to prevent out-of-sync reclosure. Therefore, out-of-sync reclosure can only be caused by faulty or manipulated measurements.

5.4.2. The speed controller

Fig. 7 highlights the speed controller (*CTRL-N-1*) as the most critical node. A malfunction or a successful cyber attack on the speed controller can cause all potential system hazards. But in order to identify the relevant factors we need to understand the component layer structure of the speed controller. It is built of two concrete nodes: (i) a Raspberry Pi (*CPT-N-1*) that handles the IP based traffic from the feedback as well as the control logic and (ii) a Digital2Analog converter that takes the digital control signal to the prime mover controller (*CPT-N-2*) and transfers it to an analog signal between 0 V and 5 V. The devices are hard-wired over USB (*CPT-C-1*). From the hazards in Fig. 7, out-of-sync reclosure (*H 1*), violation of power quality (*H 3*) and inability to meet local demand (*H 5*) can be caused by *CPT-N-1*. Only the operational limits of the generator set need to be ensured by *CPT-N-2*. Further, *CPT-N-2* is not connected to any network directly. Therefore its integrity in the cyber domain depends on the integrity of *CPT-N-1*. This shows that the reliability and the integrity of *CPT-N-1* should have highest priority among all devices in the system.

5.4.3. Wide area connection between remote PMU and controller

The final case that we investigate in detail is the connection between the remote host-based PMU and the speed controller (*CTRL-C-5*). Under the assumption that we already addressed the possibility that the speed controller is compromised in the previous section, we can now focus on the communication. Fig. 7 shows that the incorrect behavior of the communication link can cause out-of-sync reclosure, power quality violations and failure to meet local demand. However, power quality violations and failure to meet local demand can, in this case, only be caused as a consequence of out-of-sync reclosure. The local speed controller would not cause these hazards solely based on the feedback from the host grid; local feedback and local stability would take priority. Therefore, only one hazard needs to be investigated in detail: *H-1*.

The component layer (Fig. 6) highlights the complexity of the connection. Traffic from the PMU is routed through a wide area network to the local network of the microgrid (*CPT-C-7*). There it has to pass through a firewall (*CPT-N-8*) to a network switch (*CPT-N-7*) before it can reach the controller. Again, Tables 1 and 2 provide the required mapping. However, the network components in the WAN are not controlled by the system architects. It is therefore not possible to ensure any safety or security constraints on this network (neither on the network nodes, nor on the physical layer). Table 8 shows which constraints have to be ensured at each component. Since there is only one system hazard (*H-1*) to ensure, the table structure deviates from the structure in Table 7.

To ensure reliability for the WAN connection (*CPT-C-7*), service level agreements (SLAs) need to be in place between the operators of the WAN and the grid operators. Availability constraints in the cyber domain (*CSTR-A-1 - CSTR-A-4*) are also covered by these SLAs for *CPT-C-7*, as are *CSTR-I-6* (measurements dropped) and *CSTR-I-8* (measurements delayed). The only aspect of *CPT-C-7* with respect to data integrity that is still controlled by the grid operator is the communication protocol (see Table 1). To ensure message integrity (*CSTR-I-5* and *CSTR-I-7*) at *CPT-C-7* we need to place integrity checks at protocol level. These can be end-to-end encryption (either at the application layer, or by using a Virtual Private Network (VPN)) or digital signatures. In Table 8 the constraints on *CPT-C-7* are split into those that need to be ensured by SLAs and those that need to be ensured at the application layer of the communication protocol.

In the internal network, a wider approach is needed. Mitigation strategies on the network switch (*CPT-N-7*) and the firewall (*CPT-N-8*) need to be balanced. Purely based on *CTRL-C-5*, two constraints can be placed at the firewall (*CPT-N-8*). First, valid traffic from *CPT-N-9* needs to be allowed to pass into the local network. Second, the firewall needs to block malicious traffic that targets the availability or integrity of the network switch (*CPT-N-7*). The network switch should — to any extent possible — ensure message integrity (*CSTR-I-5 - CSTR-I-8*) and communication availability (*CSTR-A-1 - CSTR-I-4*). Based on the open structure of typical network protocols (e.g. ARP, DNS) this might not be possible for all constraints.

5.5. Define hazard scenarios

In the previous section we highlighted how safety and security constraints are mapped between control and component layer. To finally connect all the artifacts generated in the analysis, hazard scenarios need to be described. A hazard scenario is a textual representation of one specific case during system operation in which a system flaw can lead to a hazardous control action and subsequently to a system hazard with the possibility of a system loss. Each scenario is linked to a set of violated constraints that trigger system flaws and lead to a hazardous control action being taken. The textual representation is very important. During the analysis a lot of data are generated and processed by the team of analysts but it is not easy to structure. This makes it very difficult for external personnel to understand and use the analysis results. A hierarchically structured list of hazard scenarios provides a textual representation of the final analysis results. Reading through the scenarios, a person external to the analyst team can identify the most relevant aspects and find the more detailed results from there. Therefore, it is essential to correlate the scenarios with system flaws and violated control actions.

The list of hazard scenarios can be very long and should be hierarchically structured. One generic scenario can be defined for each safety or security flaw of the system. Each scenario can then be refined iteratively into sub-scenarios. The set of scenario refinements is represented as a tree. The scenarios highlight the ways in which the system can fail. They provide the structured summary of the analysis results and the starting point for in-depth analysis. Further, they can be used to evaluate mitigation strategies for ef-

Table 8
CTRL-C-5 can cause out-of-sync reclosure (*H-1*). Mapping the relevant constrants to the component level, full circles indicate that the constraints need to be ensured at a certain component. Half circles indicate that the constraints can be affected by the component, but mitigation strategies at the specific component are not sufficient. This does not mean that mitigation at this point should be ignored, but that the effectiveness needs to be considered with care.

CSTR	SLAs	Appl. layer	*CPT-N-8*	*CPT-N-7*
I-5	○	●	◐	◐
I-6	●	○	◐	◐
I-7	○	●	◐	◐
I-8	●	○	◐	◐
A-1	●	○	◐	◐
A-2	●	○	●	◐
A-3	●	○	●	◐
A-4	●	○	●	◐
Reliab.	●	○	○	●

fectiveness. A tree of scenarios is effectively mitigated if every path from a leaf node to the root node is prevented. The system can be considered safe and secure, if scenario tree is effectively mitigated.

For space reasons it is not possible to provide a full list of all hazard scenarios at this point in the paper. However, one example is discussed in detail.

Scenario 1: The controller incorrectly believes that the voltage difference is within the limits.

Hazards: *H-1*, *H-3*, *H-5*
System Flaw: *F-1*
Hazardous Control Action: *HC-1*
Control Level Components: *CTRL-N-1*, *CTRL-N-4*, *CTRL-C-4*, *CTRL-N-5*, *CTRL-C-5*
Component Level Components: *CPT-N-1*, *CPT-N-5*, *CPT-C-6*, *CPT-N-7*, *CPT-N-8*, *CPT-N-9*, *CPT-C-7*, *CPT-C-9*

Scenario 1 represents flaw *F1* directly. Based on this scenario, the detailed analysis of the safety and security constraints will be used to create more detailed sub-scenarios.

Scenario 1.1: The speed controller (CPT-N-1) interprets the correct feedback incorrectly.

Control Level Components: *CTRL-N-1*
Component Level Components: *CPT-N-1*
Safety Constraints: Reliability of device and algorithm, Correctness of algorithm
Security Constraints: *CSTR-I-5*, *CSTR-I-7*

Scenario 1.2: The feedback received at CTRL-N-1 from the remote PMU (CTRL-N-5) is incorrect but considered valid.

Control Level Components: *CTRL-N-1*, *CTRL-N-5*, *CTRL-C- 5*
Component Level Components: *CPT-N-1*, *CPT-N-9*, *CPT-N-7*, *CPT-N-8*, *CPT-C-7*, *CPT-C-9*
Safety Constraints: Reliability of *CPT-N-9*
Security Constraints: *CSTR-I-5*, *CSTR-I-7*

Scenario 1.2.1: CTRL-N-4 sends incorrect feedback.

Control Level Components: *CTRL-N-5*
Component Level Components: *CPT-N-9*
Safety Constraints: Reliability of *CPT-N-9*
Security Constraints: *CSTR-I-5*, *CSTR-I-7*, successful exploit of *CPT-N-9*

Scenario 1.2.2: Correct feedback from CTRL-N-4 is altered or additional feedback is injected at CTRL-C-5. The communication is valid at CPT-C-1.

Control Level Components: *CTRL-C-5*
Component Level Components: *CPT-C-7*, *CPT-N-7*, *CPT-N-8*
Safety Constraints: none
Security Constraints: *CSTR-I-5*, *CSTR-I-7*

Scenario 1.2.3: Correct feedback from CTRL-N-4 is altered or additional feedback is injected at CTRL-C-5. The communication is invalid at CPT-C-1 but accepted.

Control Level Components: *CTRL-N-1*, *CTRL-C-5*
Component Level Components: *CPT-N-1*, *CPT-C-7*, *CPT-N-7*, *CPT-N-8*
Safety Constraints: none
Security Constraints: *CSTR-I-5*, *CSTR-I-7*

Scenario 1.3 is close to equivalent with *Scenario 1.2*. We would only need to consider the local PMU and its communication to the speed controller instead of the remote PMU. Similar considerations result in *Scenario 2* and *Scenario 3* based on *Scenario 1*. For the final three system flaws, the scenarios are designed in the same manner. The hierarchical approach is used to refine the scenarios and make the cases more concrete. Further refinement is possible for *Scenario 1.2.1* for example — ultimately we can define a scenario for each way a constraint can be violated at a specific concrete component — but a reasonable middle ground should be achieved.

6. Discussion

With our analysis we are able to describe the various ways a system can fail. Each scenario described in Sect. 5.5 highlights the safety and security constraints that are violated in order to cause a specific system hazard. The tree structure has multiple benefits.

(i) The root scenario can be used to communicate the need for mitigation strategies at board levels. They describe the faulty or malicious system behavior at high level and point out the potential system losses and thus the cost in the worst case.
(ii) Mitigation strategies can be applied at each node of the tree. All scenarios in a specific tree are effectively mitigated when every path from a child node to the root is mitigated at least at one point.
(iii) Each child node can be further refined to explicitly highlight a specific way to violate a constraint. That way attack trees can be used to further refine hazard scenarios in the cyber-security domain. If all paths through the attack tree are secured, the scenario is mitigated.
(iv) With the increased visibility that the tree structure provides the requirements for in-depth security analysis are highlighted. If a specific hazard scenario does not provide enough detail to design an effective mitigation strategy and no parent scenario can be mitigated, detailed analysis is required.

As a system-based hazard analysis technique, STPA-SafeSec does not directly provide quantifiable results. However, known approaches to make system-based hazard analysis techniques like HAZOP more quantifiable can also be applied to STPA-SafeSec. Further, methods to quantify attack trees can also be applied to the tree structure of STPA-SafeSec results.

The following examples show how the hazards caused by Scenario 1 can be mitigated. One way would be the introduction of an additional safety device at the circuit breaker. This device is hardwired to both sides of the circuit breaker and locally measures voltage magnitude, frequency and phase angle. If the difference over the circuit breaker in any of the metrics is too high, the device prevents the circuit breaker from closing. This strategy would mitigate *H-1* (out-of-sync reclosure) and subsequently also *H-3* and *H-5* in Scenario 1. Thus, all hazards of the scenario would be mitigated and the child scenarios do not need further consideration. This approach has the drawback that a set of devices might be compromised without the operator's notice, as the safety device shields the feedback. The problem here is with attribution and state awareness. Additional monitoring capabilities need to be installed to achieve better state awareness of the ICT systems. However, STPA-SafeSec focuses on the loss free operation of the system: Thus, it does not identify the need for these systems necessarily if this sort of state awareness is not required to mitigate a certain hazard scenario.

A different approach would be to consider the sub-scenarios in detail. Scenario 1.2.2 can be mitigated with the use of an application layer protocol that ensures data integrity (either with the use of cryptographic signatures or end-to-end encryption). Scenario 1.2.3 requires that the data integrity checks are actually validated at *CPT-N-1*. Mitigation of Scenario 1.2.1 is more difficult. The reliability constraint can be ensured over device quality of *CTRL-N-4*, or redundancy. But to ensure the integrity of the device, we need an in-depth security analysis of the device. The same in-depth analysis is required for *CPT-N-1* to mitigate Scenario 1.1.

Based on the two examples, it seems obvious that the first strategy is more effective. First, one additional device is most probably more cost effective than the detailed security analysis, the change in protocols and the mitigation strategies that need to be applied as a result of the further analysis. But Scenario 1 is only one hazard scenario in the system. An analysis of scenarios based on system flaw *F-5* (the speed controller incorrectly believes that no set-point change is required) cannot be mitigated in the same way. Incorrect feedback to the speed controller will indefinitely prevent the reclosure of the circuit breaker since no synchronization can be achieved. To mitigate these scenarios integrity checks at application level are necessary as well as the in-depth analysis of *CPT-N-1*. Under these considerations the more effective mitigation would be to ensure the security constraints.

This shows how the results of STPA-SafeSec can be used to weigh complex decisions about the most critical components and the most effective mitigation strategies. Further, STPA-SafeSec highlights scenarios where no physical mitigation strategy is valid. To limit the damage a priority can be given to system losses in order to prevent the most critical losses by accepting less critical losses. In our example it would be possible to accept *L-4* (interruption of power supply to consumer loads) to prevent *L-1* (injury to humans).

The presented analysis does not present automated ways to perform STPA-SafeSec. However, semi-automated methods that were defined for traditional STPA can be directly applied to some parts of STPA-SafeSec. Additionally, similar semi-automated methods can be applied for the novel steps in STPA-SafeSec.

7. Conclusion

In this paper, we have analyzed safety and security aspects of a microgrid use case around synchronous-islanded operation. To control these operational states, an increased integration of power systems with ICT communication is needed. This integration motivates new approaches to analyze systems with respect to safety and security. We have presented such a novel approach — STPA-SafeSec — that unifies and extends the System Theoretic Process Analysis (STPA) technique for safety analysis and STPA-sec for security analysis. The novel contribution of this work is to formalize an approach to analyze the dependencies between physical limitations imposed by real power equipment and the capabilities of an attacker in the cyber domain. This formalization allows analysts to use previous research results in the security domain (e.g., Wang and Lu, 2013) and apply them to an infrastructure. Further, known methods to quantify results from system-based hazard analysis techniques can be applied to STPA-SafeSec. We are able to show that safety and security constraints need to be tackled together to identify the full set of scenarios that lead to system losses. With the introduction of a component layer, we were able to identify the physical impact that existing security vulnerabilities like GPS jamming (Hu and Wei, 2009) can have on the system. The ability to highlight the physical effects of security vulnerabilities or system flaws based on high level system losses make the results easier to communicate at board level than generic security analysis result provided, for example, by Dondossola et al. (2008). STPA-SafeSec is further able to guide in-depth security analysis to the most critical components and integrate the results. We showed an example where different mitigation strategies from the safety and security domain mitigate a scenario that lead to high level system losses. Furthermore, the results from STPA-SafeSec can be used to design complex reactive frameworks that ensure system safety, such as the one presented by Ten et al. (2010).

Acknowledgements

This work was partly funded by the EPSRC CAPRICA (Contract No. EP/M002837/1) project and the EU FP7 SPARKS project (Contract No. 608224).

References

Avizienis, A, Laprie, J-C, Randell, B, Landwehr, C, 2004. Basic concepts and taxonomy of dependable and secure computing. Dependable Secure Comput IEEE Trans 1 (1), 11–33.

Best, R, Morrow, D, Laverty, D, Crossley, P, 2008. Universal application of synchronous islanded operation.

Caldon, R, Rossetto, F, Turri, R, 2004. Temporary islanded operation of dispersed generation on distribution networks. 39th International Universities Power Engineering Conference, 2004. UPEC 2004, vol. 3, pp. 987–991 vol. 2..

Considine, T, Cox, W, Cazalet, EG, 2012. Understanding micrograds as the essential architecture of smart energy. Grid Interop Forum, Texas.

Dondossola, G, Szanto, J, Masera, M, Nai Fovino, I, 2008. Effects of intentional threats to power substation control systems. Int J Crit Infrastruct 4 (1), 129–143.

Duckworth, HA, Moore, RA, 2010. Social responsibility: failure mode effects and analysis. CRC Press.

Dunjó, J, Fthenakis, V, Vlchez, JA, Arnaldos, J, 2010. Hazard and operability (HAZOP) analysis. A literature review. J Hazard Mater 173 (1–3), 19–32.

Farhangi, H, 2010. The path of the smart grid. Power Energy Mag IEEE 8 (1), 18–28.

Friedberg, I, Laverty, D, McLaughlin, K, Smith, P, 2015. A cyber-physical security analysis of synchronous-islanded microgrid operation. 3rd International Symposium for ICS & SCADA Cyber Security Research 2015 (ICS-CSR 2015).

Hu, H, Wei, N, 2009. A study of GPS jamming and anti-jamming. Power Electronics and Intelligent Transportation System (PEITS), 2009 2nd International Conference on, vol. 1, pp. 388–391.

Kang, B, Maynard, P, McLaughlin, K, Sezer, S, Andren, F, Seitl, C, et al., 2015. Investigating cyber-physical attacks against IEC 61850 photovoltaic inverter installations. Emerging Technologies Factory Automation (ETFA), 2015 IEEE 20th Conference on, pp. 1–8.

Karnouskos, S, 2011. Stuxnet worm impact on industrial cyber-physical system security. IECON 2011 — 37th Annual Conference on IEEE Industrial Electronics Society.

Kordy, B, Piètre-Cambacédès, L, Schweitzer, P, 2014. DAG-based attack and defense modeling: don't miss the forest for the attack trees. Comput Sci Rev 13–14, 1–38.

Kundur, D, Feng, X, Mashayekh, S, Liu, S, Zourntos, T, Butler-Purry, KL, 2011. Towards modelling the impact of cyber attacks on a smart grid. Int J Secur Netw 6 (1), 2–13.

Laprie, J-C, Kanoun, K, Kaâniche, M, 2007. Modelling interdependencies between the electricity and information infrastructures. Computer safety, reliability, and security. Springer, pp. 54–67.

Laverty, DM, Morrow, DJ, Best, R, Crossley, PA, 2008. Internet based phasor measurement system for phase control of synchronous islands. 2008 IEEE Power and Energy Society General Meeting — conversion and delivery of electrical energy in the 21st century. IEEE, pp. 1–6.

Lawley, H, 1974. Operability studies and hazard analysis. Chem Eng Process 70 (4), 45–56.

Lee, R, Assante, M, Connway, T, 2014. ICS CP/PE (Cyber-to-Physical or Process Effects) case study paper — German steel mill cyber attack Technical report, SANS ICS.

Leveson, N, 2004. A new accident model for engineering safer systems. Saf Sci 42 (4), 237–270.

Leveson, NG, 2011. Engineering a safer world. Systems Thinking Applied to Safety, the MIT Press, Cambridge, MA, USA.

Salter, C, Saydjari, OS, Schneier, B, Wallner, J, 1998. Toward a secure system engineering methodology. In: Proceedings of the 1998 Workshop on New Security Paradigms. ACM, pp. 2–10.

Sridhar, S, Hahn, A, Govindarasu, M, 2012. Cyber physical system security for the electric power grid. P IEEE 100 (1), 210–224.

Srivastava, A, Morris, T, Ernster, T, Vellaithurai, C, Pan, S, Adhikari, U, 2013. Modeling cyber-physical vulnerability of the smart grid with incomplete information. IEEE Trans Smart Grid 4 (1), 235–244.

Standard, M, 1980. Procedures for performing a failure mode, effects and criticality analysis MIL-STD-1629, November, AMSC Number N3074.

Ten, C-W, Liu, C-C, Manimaran, G, 2008. Vulnerability assessment of cybersecurity for SCADA systems. Power Syst IEEE Trans 23 (4), 1836–1846.

Ten, C-W, Manimaran, G, Liu, C-C, 2010. Cybersecurity for critical infrastructures: attack and defense modeling. Syst Man Cybern Part A Syst Hum IEEE Trans 40 (4), 853–865.

Thomas, IVJP, 2013. Extending and automating a systems-theoretic hazard analysis for requirements generation and analysis Ph.D. thesis; Massachusetts Institute of Technology.

Vesely, WE, Goldberg, FF, Roberts, NH, Haasl, DF, 1981. Fault tree handbook Technical report, DTIC Document.

Wang, W, Lu, Z, 2013. Cyber security in the Smart Grid: survey and challenges. Comput Netw 57 (5), 1344–1371.

Watson, H, 1961. Launch control safety study Technical report, Bell Laboratories: Murray Hill, NJ.

Weiss, JD, 1991. A system security engineering process. In: Proceedings of the 14th National Computer Security Conference, vol. 249.

Young, W, Leveson, N, 2013. Systems thinking for safety and security. In: Proceedings of the 29th Annual Computer Security Applications Conference on — ACSAC ‘13. ACM Press, New York, New York, USA., pp. 1–8.

Young, W, Leveson, NG, 2014. An integrated approach to safety and security based on systems theory. Commun ACM 57 (2), 31–35.

Zhang, Z, Gong, S, Dimitrovski, AD, Li, H, 2013. Time synchronization attack in smart grid: impact and analysis. Smart Grid IEEE Trans 4 (1), 87–98.