

Cyber LOPA: An Integrated Approach for the Design of Dependable and Secure Cyber-Physical Systems

Ashraf Tantawy , Member, IEEE, Sherif Abdelwahed, Senior Member, IEEE,
and Abdelkarim Erradi , Member, IEEE

Abstract—Safety risk assessment is an essential process to ensure a dependable cyber-physical system (CPS) design. Traditional risk assessment considers only physical failures. For modern CPSs, failures caused by cyber attacks are on the rise. The focus of latest research effort is on safety–security lifecycle integration and the expansion of modeling formalisms for risk assessment to incorporate security failures. The interaction between safety and security lifecycles and its impact on the overall system design, as well as the reliability loss resulting from ignoring security failures, are some of the overlooked research questions. This article addresses these research questions by presenting a new safety design method named cyber layer of protection analysis (CLOPA) that extends the existing layer of protection analysis (LOPA) framework to include failures caused by cyber attacks. The proposed method provides a rigorous mathematical formulation that expresses quantitatively the tradeoff between designing a highly reliable and a highly secure CPS. We further propose a co-design lifecycle process that integrates the safety and security risk assessment processes. We evaluate the proposed CLOPA approach and the integrated lifecycle on a practical case study of a process reactor controlled by an industrial control testbed and provide a comparison between the proposed CLOPA and current LOPA risk assessment practice.

Index Terms—Cyber-physical system (CPS), hazard and operability (HAZOP), IEC 61511, layer of protection analysis (LOPA), NIST SP 800-30, risk assessment, supervisory control and data acquisition, safety instrumented system (SIS), safety integrity level (SIL), security.

I. INTRODUCTION

A CYBER physical system (CPS) is an integration of a physical process with computation and networking required for physical system monitoring and control. The integration of process dynamics with those of computation and networking brings

Manuscript received February 24, 2021; revised September 22, 2021 and January 16, 2022; accepted March 11, 2022. Date of publication April 22, 2022; date of current version June 2, 2022. This work was supported by the Qatar National Research Fund (a member of the Qatar Foundation) under Grant NPRP 9-005-1-002. The statements made herein are solely the responsibility of the authors. Associate Editor: W. Dong. (*Corresponding author: Ashraf Tantawy.*)

Ashraf Tantawy is with the School of Computer Science and Informatics, De Montfort University, LE1 9BH Leicester, U.K. (e-mail: ashraf.tantawy@dmu.ac.uk).

Sherif Abdelwahed is with the Department of Electrical and Computer Engineering, Virginia Commonwealth University, Richmond, VA 23284 USA (e-mail: sabdelwahed@vcu.edu).

Abdelkarim Erradi is with the Department of Computer Science and Engineering, Qatar University, Doha 2713, Qatar (e-mail: erradi@qu.edu.qa).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TR.2022.3163652>.

Digital Object Identifier 10.1109/TR.2022.3163652

a plethora of engineering challenges. As the majority of CPSs are deployed in mission-critical applications, the dependability and resilience to failures is a key design property for modern CPS.

To ensure that a given CPS is dependable, a risk assessment is carried out at both design time and operation time. The risk assessment process highlights the system weaknesses and helps define the safety requirements that need to be met to achieve the target reliability measures. The classical approach to perform the risk assessment is to consider physical failures only. As state-of-the-art CPS designs move to open-source hardware and software, cyber attacks have become a source of failure that cannot be ignored.

Realizing the critical nature of CPS cyber attacks and their impact on the safety of people and environment, as well as the potential catastrophic financial losses, the research community developed several approaches to integrate security aspects into the safety risk assessment process. This integration has been done mainly by extending the reliability modeling formalism to incorporate security-related risks. One of the overlooked research questions is how safety and security interact with each other and how this interaction would impact the overall system design. Putting this research question in a different format: *Is there a tradeoff between designing a highly reliable and a highly secure system?* a related research question is: *If we ignore the cyber security attacks in the design process, what is the impact on the overall system reliability? Is the reliability gain worth the complexity introduced by integrating security both at design and runtime?* A follow-up research question is: *Under what conditions can we ignore security failures?*

In order to better understand the interaction between safety and security lifecycles in the system design process, we consider in this article the safety risk assessment process and study the impact of overlooking failures caused by cyber attacks. We refer to such failures as security failures in the rest of this article. By formally introducing the failures caused by attacks into the risk assessment process, we can define the reliability requirements for the cyber components of the system as a function of both the failure rate of physical components and the resilience to cyber attacks. This formal requirement specification enables us to understand the design tradeoff between higher reliability of physical components versus higher resilience of cyber components, and the sensitivity of the overall system performance to both types of failures. In addition, we can gain insight into the interplay between safety and security and how to integrate both lifecycles

during the design process. More specifically, we consider the layer of protection analysis (LOPA), a widely adopted risk assessment method that follows a hazard identification study, such as hazard and operability (HAZOP). LOPA is carried out to identify whether an additional safety instrumented system (SIS) is needed for specific hazardous scenarios to achieve the target risk level. As a modern SIS is typically an embedded device, it has both physical and security failure modes. We mathematically derive the SIS design constraints in terms of both physical and security failure probabilities. Additionally, we propose an integrated safety–security design process that shows the flow of information between both lifecycles.

We can classify the research work on combining safety and security for CPSs into two broad categories that try to answer the following research questions: 1) given the independent safety and security lifecycles, what are the similarities/differences and how could the two lifecycles be aligned or unified? This research direction usually focuses on answering the question “what to do,” rather than “how to do it”; and 2) for a given CPS, how can we carry out risk assessment (qualitative/quantitative) that considers both physical failures and cyber attacks? Consequently, how can we unify the process of safety and security requirements definition and verification? This research direction focuses on common modeling techniques that can incorporate both safety and security failures and often extends model-based engineering body of knowledge and tools to incorporate security requirements in the design process. In Section VI, we survey the main results for each research direction. A more thorough survey is presented in [1] and [2].

Our contribution: The work presented in this article addresses both research directions with a new approach. First, we integrate both safety and security lifecycles based on a rigorous mathematical formulation that captures their interaction. The formulation enables the designer to assess how a security design decision would impact system safety. This is in contrast to the existing research work that does not explicitly model the dynamic safety–security lifecycle interaction. Second, we develop an integrated safety–security design lifecycle and show in detail how to apply it to a real-world design, distinguishing the work from abstract research on risk assessment that does not carry over to the design stage. Finally, our approach is founded on LOPA, a practical approach that is extensively used in industry, giving the approach the merit for industrial implementation.

The rest of this article is organized as follows. Section II introduces the background information required for problem setup, including IEC 61511 safety lifecycle and the LOPA method, cyber dependence between control and safety systems, and cyber security risk assessment. Section III proposes a new LOPA mathematical formulation called cyber layer of protection analysis (CLOPA) that incorporates failures due to cyber attacks. Section IV proposes an integrated safety–security lifecycle process. Section V presents a case study for the design of a safety system for a chemical reactor, comparing classical LOPA approach to the proposed CLOPA formulation. Section VI summarizes the related work on safety–security co-design. Finally, Section VII concludes this article.

II. SAFETY AND SECURITY RISK ASSESSMENT

There are two main embedded systems that control and safeguard a given physical system: the control system and the safety system. In the process industry, the control system is referred to as the basic process control system (BPCS), and the safety system is referred to as the SIS. In practice, both the systems typically have a programmable controller architecture with one or more back planes, processor cards, and a variety of input–output interface cards [3]. For larger systems, BPCS and SIS architectures comprise multiple distributed nodes connected via a communication backbone. Fig. 2 depicts the two systems and their connectivity over a control network. In the following, we briefly discuss the SIS design lifecycle, BPCS and SIS security lifecycles, and their interaction.

A. IEC 61511 Safety Lifecycle Process

Fig. 1 shows the SIS design lifecycle according to IEC 61511 standard [4]. The design starts with hazard and risk assessment, where systems hazards are identified. HAZOP study, what if analysis, and fault tree analysis are the most common methods at this stage [5]. The risk assessment phase ranks each identified risk according to its likelihood and consequence, either quantitatively or qualitatively, and associates a risk ranking for each hazard. The resulting list of hazards and associated risk ranking is used as an input to the second phase focused on the allocation of safety functions to protection layers. This phase deals only with hazards that exceed a threshold risk rank that an organization is willing to accept. For each hazardous scenario, there is a target mitigated event likelihood (TMEL) measure that is defined based on the risk rank. The purpose of this phase is to check if the TMEL is met with existing protection layers. If not, an additional protection layer is recommended, often in the form of a new safety instrumented function (SIF) with a predefined safety integrity level (SIL) to cover the gap to the TMEL. The SIF comprises one or more sensors, a logic solver, and one or more actuators. The logic solver is commonly referred to as the SIS. An example SIF is illustrated in Fig. 6 for an overflow hazardous scenario of a reactor system, which will be discussed in detail in Section V. Risk matrix, risk graph, and LOPA are the most commonly used methods for the allocation of safety functions to protection layers [3].

The third phase is the development of the safety requirement specification (SRS), which documents all the functional and timing requirements for each SIF. The fourth phase is the detailed design and engineering. Phases 5–8 are concerned with system installation and commissioning, operation, modification, and decommissioning. Phase 2 is where the CPS control and safety systems are considered in the risk assessment process. Therefore, we study this phase in depth in this article. Since LOPA is the predominant approach for this phase, we limit our discussion to LOPA methodology. Other approaches could be adopted in a similar way.

The underlying assumption in LOPA analysis is that all protection layers, including the new SIF, are independent. In other words, if one layer failed, this does not increase or decrease



Fig. 1. IEC 61511 SIS design lifecycle (adopted from [4]).

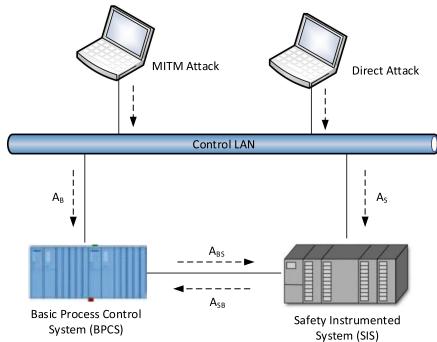


Fig. 2. Snapshot of an industrial control system architecture showing BPCS–SIS connectivity and potential attack vectors.

the likelihood of failure of the other layers. This assumption simplifies the mathematical analysis significantly, as it allows the multiplication of individual probabilities to obtain the required joint probability. The simplicity of LOPA calculations is probably one of the key reasons behind its widespread adoption by industry. Unfortunately, when cyber security is considered as a potential failure in LOPA analysis, the independence assumption between the control and safety systems no longer holds, as explained in the next section.

B. Control and Safety System Cyber Dependence

Each of the control and safety systems has two modes of failure: BPCS physical failures, B_p , BPCS security failure, B_c , SIS physical failure, S_p , and SIS security failure S_c . For physical failures, IEC 61511 standard strongly recommends complete separation between the control and safety systems of any plant. This separation includes sensors, computing devices, and final elements such as valves and motors. Separation also includes any common utility such as power supplies. The industry adopted this separation principle; hence, BPCS and SIS physical failures could be accurately assumed to be independent, i.e., $P[B_p, S_p] = P[B_p]P[S_p]$.

One exception to the separation between BPCS and SIS is the cyber communication link between the control and safety systems. Fig. 2 shows a snapshot of a typical industrial control system architecture showing the communication link between the BPCS and the SIS. BPCS–SIS communication could be over the control LAN or via a dedicated point to point serial link. The communication protocol is typically an open standard such as Modbus or DNP3 [6], [7]. This type of communication exists in many industrial installations to exchange plant data, as the data from field devices connected to the safety system are not accessible from the BPCS and *vice versa*. Given this architecture, we can define two attack vectors for SIS compromise: 1) a direct attack that exploits an existing controller vulnerability could be

launched against the SIS node. This could be via any node on the control LAN or using man-in-the-middle (MITM) attack that exploits the BPCS–SIS communication. We designate this attack event by A_S in Fig. 2; and 2) by compromising the BPCS first and then exploiting the BPCS–SIS link to compromise the SIS. We designate this pivot attack by the sequence of events A_B and A_{BS} in Fig. 2. Furthermore, we designate the attack event from the SIS to the BPCS by A_{SB} . The attack sequence $A_B \rightarrow A_{BS}$ may be easier if the SIS is highly secured such that a direct attack may be infeasible. This is particularly true if we consider the fact that the BPCS is a trusted node to the SIS.

The above analysis shows a clear dependence between the control and safety systems that violates the original LOPA independence assumption. The security failures for the BPCS and the SIS are no longer independent because of the data communication coupling. We can formulate the different security failure probabilities as in (1)–(3) using basic probability laws with the aid of Fig. 2, where $P[A_i]$ is interpreted as the probability of success of attack A_i . Furthermore, the considered attack A_i should have the impact of stopping the BPCS or SIS from performing its intended control or safeguard function as related to the hazard under study. This is important because not all attacks that exploit controller vulnerabilities result in a process hazard. Therefore, the attacks considered represent a subset of the complete set of attacks that could exploit the BPCS or SIS vulnerabilities. Accordingly, from hereafter, $P[A_B]$, $P[A_S]$, $P[A_{BS}]$, and $P[A_{SB}]$ refer to the relevant attacks that cause a process hazard. This concept is revisited throughout this article and is made more clear in the case study when sample attacks are presented. For a case study example on the fact that not all cyber failures have a system reliability consequence, we refer the interested reader to [8] for a study on the impact of different software failure modes on system reliability for the electric power grid domain. Finally, it can be easily shown that if the BPCS–SIS communication link does not exist, or fully secured, then $P[A_{SB}] = P[A_{BS}] = 0$, and (3) reduces to the independent case $P[S_c, B_c] = P[S_c]P[B_c]$

$$P[B_c] = P[A_B] + P[A_S]P[A_{SB}] - P[A_B]P[A_S]P[A_{SB}] \quad (1)$$

$$P[S_c] = P[A_S] + P[A_B]P[A_{BS}] - P[A_B]P[A_S]P[A_{BS}] \quad (2)$$

$$P[S_c, B_c] = P[A_B](P[A_S] + P[A_{BS}]) + P[A_S]P[A_{SB}] - P[A_B]P[A_S](P[A_{BS}] + P[A_{SB}]). \quad (3)$$

C. Cyber Security Risk Assessment

The calculation of the probability of cyber attacks A_S , A_B , and A_{BS} could be performed during the cyber security risk assessment process. This requires a detailed specification of

the BPCS and the SIS and their connectivity, including the embedded system hardware, operating system, running software services, and the network connectivity. According to the National Institute of Standards and Technology (NIST) SP 800-30 standard, “Guide for Conducting Risk Assessments,” the cyber security lifecycle process stages are: 1) asset identification, where the particular cyber components and their criticality levels are identified; 2) vulnerability identification, along with the associated threats and attack vectors; 3) the development of relevant attack trees for each attack scenario identified; 4) penetration testing to validate the vulnerability findings and attack scenarios and to help estimating the effort and probability for individual attack steps for each scenario; and 5) risk assessment to identify the scenarios with unacceptable risk [9]. Fig. 5 shows the BPCS and SIS cyber security lifecycles. In this article, we follow the same cyber security lifecycle, but with the physical process as the main focus. Therefore, for asset identification, the cyber component criticality is primarily identified by its failure impact on the operation of the connected physical component. Similarly, for vulnerability identification, threats and attack vectors are filtered by their impact on the physical process. Attacks that do not disturb the controlled process are ignored as they have no direct impact on the process safety. In addition, such impacts take place with much higher probability at the corporate network level, so they can be ignored with minimum impact on the risk assessment at the control network level. For more detailed discussion on process-driven attack identification, we refer the reader to [10].

For the presented architecture, the BPCS and the SIS are the critical components in direct contact with the process. The calculation of the required BPCS and SIS security failure probabilities could be typically carried out with the aid of attack trees [11]. The attack tree enumerates all the possible routes to compromise the system, and each edge is assigned a probability representing the likelihood of the associated event. Using basic probability laws, the overall probability of a system compromise could be calculated. Section V presents an example of such calculation. We re-emphasize the fact that the scope of cyber security risk assessment and attack trees in this case will be limited to attacks targeting the physical process to cause a process hazard. Although information security attacks with objectives such as stealing information are possible, most of this information is already available at the corporate network level, and an attacker who penetrates down to the control network level to compromise an BPCS or SIS will conceivably have the goal of physical process attack.

III. CLOPA: LOPA WITH SECURITY FAILURES

A. Mathematical Formulation

In risk assessment, an initiating event is an unplanned event that when occurring may lead to a hazard. Examples of initiating events include equipment failure, human error, and cyber attacks. A system hazard will take place if one or more of the initiating events occur, and all the associated protection layers against that hazard fail simultaneously. The main objective of LOPA is to calculate the expected number of hazardous events

per time interval and compare it to the TMEL. We designate the random variable representing the number of events per unit time for a specific initiating event by N , the random variable representing the simultaneous failure of protection layers when the initiating event occurs by L , where L is Bernoulli distributed with success probability p , and the random variable representing the number of hazards per time interval by H . We then have

$$H = \sum_{i=1}^N \mathbb{I}_E(l_i) \quad (4)$$

where \mathbb{I} is the indicator function, and the set $E = \{l : l_i = 1, i = 1:N\}$. We note that for a given $N = k$, H is a binomially distributed random variable with expected value $E[H|N = k] = kp$. Therefore

$$\begin{aligned} E[H] &= \sum_{k=0}^{\infty} E[H|N = k]P[N = k] \\ &= p \sum_{k=0}^{\infty} kP[N = k] = pE[N] = p\lambda \end{aligned} \quad (5)$$

where λ represents the expected value of the number of initiating events per unit time, N . Although N is typically modeled by a Poisson random variable in reliability engineering, we do not assume any specific distribution in the analysis. This is particularly important because some initiating events considered in the article, such as security failures, are not accurately modeled by a Poisson distribution.

Equation (5) is the underlying mathematical concept behind LOPA analysis. Essentially, for each initiating event, the likelihood λ is estimated from field data, and the probability of simultaneous failure of all protection layers is specified. Finally, the expected number of hazards per unit time, $E[H]$, considering all initiating events, is estimated and compared to the prespecified TMEL. If $E[H] > \text{TMEL}$, then a SIS is required with a probability of failure on demand $P[S_p]$ (or equivalently a risk reduction factor (RRF) = $1/P[S_p]$) that achieves $E[H] \leq \text{TMEL}$.

In order to express the LOPA formula in (5) in terms of all protection layers, including the BPCS and the SIS, we introduce some mathematical notation. We designate the set of initiating events for a given hazardous scenario by $\mathcal{I} = \{I_1, I_2, \dots, I_n, B_p, \mathcal{A}_r\}$, where n is the number of possible initiating events excluding BPCS failures, B_p denotes the BPCS physical failure event, and \mathcal{A}_r denotes the set of attacks relevant to the hazard under study. We express the associated set of event likelihoods by $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_n, \lambda_p, \lambda_r\}$. Furthermore, we denote the set of all possible protection layers by $\mathcal{L} = \{L_1, L_2, \dots, L_m\}$, where m is the number of existing protection layers, excluding the BPCS and the SIS. BPCS protection is denoted by B , and SIS protection is denoted by S . For each initiating event i , there is a subset of protection layers $\mathcal{L}_i \subseteq \mathcal{L} \cup \{B, S\}$ that could stop the propagation of a hazard from causing its consequences. Table I shows a sample LOPA table using the introduced terminology.

TABLE I
SAMPLE LOPA TABLE

Initiating Event	Likelihood λ_i (yr)	$L_1 \dots L_m$	BPCS (B)	TMEL
I_1	λ_1	$\leftarrow P[\mathcal{L}_1] \rightarrow$	$P[B]$	10^{-x}
\dots	\dots			\dots
I_n	λ_n	$\leftarrow P[\mathcal{L}_n] \rightarrow$	$P[B]$	
B_p	λ_p	$\leftarrow P[\mathcal{L}_B] \rightarrow$	1	
\mathcal{A}_r	λ_c	$\leftarrow P[\mathcal{L}_B] \rightarrow$	$P[B]$	

$P[\mathcal{L}]$ refers to the combined probability of failure of protection layers applicable to the initiating event from L_1 to L_m .

B. Semantically Relevant Attack Event Formulation

We designate the set of all possible attacks against the BPCS by \mathcal{A} . For a given hazard under consideration, the subset of attacks that would lead to this hazard, i.e., the contextually or semantically relevant attacks, is designated by \mathcal{A}_r . To estimate the likelihood of relevant cyber attacks (expected number per unit time), λ_c , we assume an attacker profile with an average rate of launching attacks per unit time λ . For every launched attack, the attacker is presented with the complete set of attacks \mathcal{A} and selects only one attack a with probability $P[A = a] = \alpha_a$ that is dependent on the attacker profile, such that

$$\sum_{a \in \mathcal{A}} P[A = a] = \sum_{a \in \mathcal{A}} \alpha_a = 1. \quad (6)$$

The likelihood of cyber attack $a \in \mathcal{A}_r$ is then $\lambda_a = \lambda \alpha_a$. This cyber attack has the potential to cause a system hazard if both the BPCS and the SIS fail jointly to stop the attack (either physical or cyber failure). We designate this probability by $P_a[S, B]$. The exact approach to include every attack $a \in \mathcal{A}_r$ in the LOPA table is to treat each attack as an individual entry, akin to the last row in Table I, with initiating event likelihood λ_a . However, this approach has two main drawbacks: First, the number of attacks could be large, and this would grow the LOPA sheet significantly. Second, the treatment of each attack individually would not allow us to utilize attack modeling techniques such as attack trees that model collectively all the possible attack paths for one attack objective. Therefore, we adopt an alternative approach, where all relevant attacks $a \in \mathcal{A}_r$ could be represented by one entry in the LOPA table. The following lemma summarizes the approximate solution. The proof is included in the Appendix.

Lemma 3.1: Assume a given hazard scenario H , a control system BPCS, an average rate of launching attacks against BPCS λ , Hazard H semantically-relevant attack set \mathcal{A}_r , and probability α_a of selecting attack $a \in \mathcal{A}_r$. Then, the impact of all initiating events $a \in \mathcal{A}_r$ on the LOPA calculation could be approximated by a single initiating event with likelihood $\lambda_c = \lambda \sum_{a \in \mathcal{A}_r} \alpha_a$ and a BPCS failure probability with respect to the combined set of attacks $a \in \mathcal{A}_r$, where each attack probability is weighted by the factor $\gamma_a = \alpha_a / \sum_{a \in \mathcal{A}_r} \alpha_a$.

The lemma enables us to use attack trees with leaf nodes weighted by γ_a to calculate the BPCS security failure probability in response to the combined set of attacks \mathcal{A}_r with likelihood $\lambda \sum_{a \in \mathcal{A}_r} \alpha_a$. For the special case where the cyber attacker profile results in random selection of the attack $a \in \mathcal{A}$, e.g., an attacker with no knowledge about the system, the likelihood

reduces to $\lambda |\mathcal{A}_r| / |\mathcal{A}|$ and the leaf node weights reduce to $\gamma_a = 1 / |\mathcal{A}_r| \forall a$. We use this special case in the case study in Section V.

C. Cyber LOPA Formulation

With the introduced notation, the expected number of hazards in (5), which should be less than the TMEL, could be expanded as

$$E[H] = P[S, B] \left(\sum_{i=1}^n (\lambda_i P[\mathcal{L}_i]) + \lambda_c P[\mathcal{L}_B] \right) \\ + \lambda_p P[S] P[\mathcal{L}_B] \leq \text{TMEL} \quad (7)$$

where \mathcal{L}_B is the set of protection layers for BPCS physical or security failure event, and we assume that all protection layers are independent of the BPCS and the SIS, while keeping the dependence between the BPCS and the SIS. In addition, higher order probability terms resulting from multiple initiating events are ignored due to their insignificance.

To calculate the joint failure probability $P[S, B]$, we use basic probability laws and the fact that the BPCS and the SIS have both the physical and cyber modes of failure, as explained in Section II-B, to obtain

$$P[S, B] = P[S_p] (P[B_p](1 - P[B_c] - P[S_c]) + P[B_c]) \\ + P[S_c, B_c] (1 - P[S_p] - P[B_p] + P[S_p]P[B_p]) \\ + P[S_c]P[B_p]. \quad (8)$$

Substituting (8) into (7), we obtain the general LOPA equation in (9). We call this expanded version of LOPA hereafter CLOPA

$$P[S_p] \leq \frac{\beta - (\alpha_1 P[S_c] + \alpha_2 P[S_c, B_c])}{\alpha_1 - \alpha_1 P[S_c] + \alpha_2 P[B_c] - \alpha_2 P[S_c, B_c]} \quad (9)$$

where

$$\alpha_1 = P[B_p] \left(\sum_{i=1}^n (\lambda_i P[\mathcal{L}_i]) + \lambda_c P[\mathcal{L}_B] \right) + \lambda_p P[\mathcal{L}_B] \quad (10)$$

$$\alpha_2 = (1 - P[B_p]) \left(\sum_{i=1}^n (\lambda_i P[\mathcal{L}_i]) + \lambda_c P[\mathcal{L}_B] \right) \quad (11)$$

$$\beta = \text{TMEL}. \quad (12)$$

In order to define the CLOPA formula in terms of the actual design variables $P[A_S]$ and $P[A_{BS}]$ that represent the probability of security failures of actual CPS components, we substitute (1)–(3) into (9) to obtain

$$P[S_p] \leq \frac{\beta - \gamma_1 P[A_S] - \gamma_2 P[A_{BS}] (1 - P[A_S])}{\gamma_3 - \gamma_3 P[A_S] - \gamma_2 P[A_{BS}] (1 - P[A_S])} \quad (13)$$

where:

$$\gamma_1 = \alpha_1 + \alpha_2 [P[A_B] + P[A_{SB}] (1 - P[A_B])] \quad (14)$$

$$\gamma_2 = (\alpha_1 + \alpha_2) P[A_B] \quad (15)$$

$$\gamma_3 = \alpha_1 + \alpha_2 P[A_B]. \quad (16)$$

Equation (13), along with (10)–(12) and (14)–(16), represents the general CLOPA formulation to design the SIS. It represents

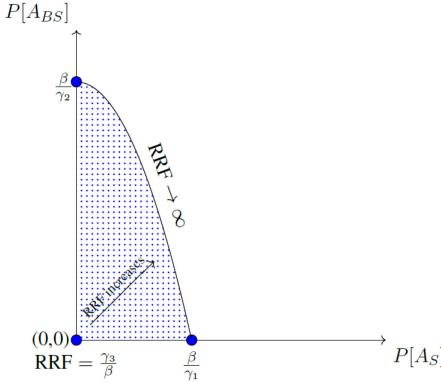


Fig. 3. CLOPA design region (shaded). Any point in the shaded region results in a feasible SIS. Points near the boundary require an SIS with a very high RRF value and, hence, difficult to obtain in practice.

an upper bound on the probability of physical failure for the safety system in terms of the security failure probabilities, showing clearly the coupling between the safety system and security system design. The design of the SIS should satisfy (13), where the design variables are $P[S_p]$, $P[A_S]$, and $P[A_{BS}]$. The rest are model parameters that are predetermined, including the BPCS failure marginal probabilities. This is because the BPCS design is independent of the SIS design and usually takes place earlier in the engineering design cycle. Note that we assume here that $P[A_{SB}]$ is a known parameter. This is because by completely defining the BPCS and its hardware and software specifications, the probability of a cyber attack compromising process safety could be estimated, even though the SIS is not yet completely defined. Table 1 in the Appendix summarizes the model variables, parameters, and how the model parameters are calculated.

It should be noted that with existing LOPA methodology, security failures are ignored, i.e., $P[A_B] = P[A_S] = P[A_{BS}] = P[A_{SB}] = 0$. Substituting these zero values into (13), we obtain the classical LOPA formulation

$$P[S_p] \leq \frac{\beta}{\alpha_1} = \frac{\text{TMEL}}{P[B_p] \sum_{i=1}^n (\lambda_i P[\mathcal{L}_i]) + P[\mathcal{L}_B](\lambda_p + \lambda_c)}. \quad (17)$$

D. Design Space

Using the fact that $P[S_p] \geq 0$ for a realizable safety system in (13), we obtain

$$\gamma_1 P[A_S] + \gamma_2 P[A_{BS}] - \gamma_2 P[A_S] P[A_{BS}] \leq \beta. \quad (18)$$

Fig. 3 shows the shaded region defined by the inequality in (18). The boundary curve is defined by (18) when the following equality holds:

$$P[A_{BS}] = \frac{\beta}{\gamma_2} \left(\frac{1 - \left(\frac{\gamma_1}{\beta} \right) P[A_S]}{1 - P[A_S]} \right). \quad (19)$$

The first-order derivative of the boundary curve is negative for $\gamma_1/\beta > 1$ and positive otherwise. Since $\gamma_1 > \beta$ to require a safety system [proof is straightforward by inspecting (10), (11)],

(14), and (17)], the boundary curve is concave as in Fig. 3. We note that any point in the shaded region results in a feasible SIS. Points on the boundary curve result in $P[S_p] = 0$ or equivalently $\text{RRF} \rightarrow \infty$. Points closer to the boundary would have high values for the RRF, requiring a very highly reliable SIS that may not be achievable in practice. Points closer to the origin result in lower RRF. It can be easily shown that the contour lines for (13), where $P[S_p] = C$, could be expressed as

$$P[A_{BS}] = \frac{C\gamma_3 - \beta}{\gamma_2(C - 1)} \left(\frac{1 - \left(\frac{C\gamma_3 - \gamma_1}{C\gamma_3 - \beta} \right) P[A_S]}{1 - P[A_S]} \right). \quad (20)$$

The contour line that represents the design boundary in Fig. 3 can be derived from (20) by setting $C = 0$.

We can extract the following information from this graph.

- 1) The maximum probability of security failure for the safety system by directed attacks is β/γ_1 . This probability results in an unrealizable safety system, as the required $\text{RRF} \rightarrow \infty$.
- 2) The maximum probability of security failure for the safety system by pivot attack via the BPCS is β/γ_2 . Likewise, this probability does not result in a realizable safety system.
- 3) Finally, the minimum value of the RRF is achieved at the origin for a perfectly secured safety system, where $P[A_S] = P[A_{BS}] = 0$, with the RRF given by

$$P[S_p]_{\max} = \frac{\beta}{\gamma_3}, \quad \text{RRF}_{\min} = \frac{\gamma_3}{\beta}. \quad (21)$$

Clearly, points outside the shaded region result in nonrealizable SIS. This result re-emphasizes the interplay between the safety and security systems of a CPS.

The design space highlights the major difference between LOPA and CLOPA. In LOPA, the SIS requirement is related to reliability in the form of the required SIL. In CLOPA, an additional requirement for the SIS is its security resilience, in the form of an upper bound on the probability of a security failure (cyber attack success), either directly or indirectly via the BPCS.

E. Classical LOPA Error

To obtain the error resulting from using classical LOPA, we subtract (17) from (13) to obtain

$$e_{\text{RRF}} = \frac{\zeta_1 + \zeta_2 P[A_S] + \zeta_3 P[A_{BS}](1 - P[A_S])}{\beta [\beta - \gamma_1 P[A_S] - \gamma_2 P[A_{BS}](1 - P[A_S])]}. \quad (22)$$

where

$$\zeta_1 = \beta(\gamma_3 - \alpha_1) = \beta\alpha_2 P[A_B] \quad (23)$$

$$\zeta_2 = \alpha_1\gamma_1 - \beta\gamma_3 \quad (24)$$

$$\zeta_3 = \gamma_2(\alpha_1 - \beta). \quad (25)$$

The minimum error occurs for a perfectly secured safety system, i.e., $P[A_S] = P[A_{BS}] = 0$:

$$\min e_{\text{RRF}} = P[A_B] \left(\frac{\alpha_2}{\beta} \right). \quad (26)$$

The error will be zero, i.e., classical LOPA result matches CLOPA, if the probability of BPCS security failure via a direct attack is zero.

IV. SAFETY-SECURITY CO-DESIGN

A. Design Process

The current industrial practice is to perform safety and security risk assessments independently, treating the physical and cyber components of a CPS as two separate entities. As illustrated in Section III, accurate safety risk assessment requires knowledge about the cyber components and their security failure probabilities. Formally, the objective is to design an SIS architecture \mathcal{A} that satisfies (13) in terms of both physical and security failure probabilities. Suppose that the architecture \mathcal{A} could be represented by a set of design variables represented by the vector \mathbf{x} . If we can relate the physical and security failure probabilities to the vector \mathbf{x} by $P[A_S] = f(x)$, $P[A_{BS}] = g(x)$, $P[S_p] = h(x)$, then we can use these functions to substitute the relevant probabilities into (13), and our design problem will be to find a set of values for the vector \mathbf{x} that satisfies the CLOPA constraint (13). Unfortunately, this design approach is not followed by industry for several reasons. First, abstracting a given architectural design \mathcal{A} into a set of design variables is a very difficult task, not to mention that these design variables have to be linked to both physical and security failures. Second, finding an exact or approximate representation of the functions $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ that relate the failure probabilities to the design variables may not be possible, as it is not always clear how a design decision would result in a higher or lower probability of failure. Finally, even if we were able to make a perfect modeling, the resulting problem to solve may turn into a discrete optimization problem that is not possible to solve in polynomial time.

Owing to these modeling limitations, the current industrial practice to design SISs (excluding cyber attacks) is to follow an iterative process and rely on engineering judgment during the design process. More precisely, the required risk reduction factor RRF_d is initially calculated; then, the engineering design proceeds to achieve RRF_d using both experience and industrial standard guidelines [4]. After the design is completed, design verification is conducted to calculate the RRF of the proposed design RRF_v . If the resulting $RRF_v \geq RRF_d$, then the design stops. Otherwise, the design is refined until the condition $RRF_v \geq RRF_d$ is satisfied. In the following, we will adopt the same iterative design approach for CLOPA.

Fig. 4 illustrates the iterative design process. We start with initial values $(P[A_S], P[A_{BS}], RRF_d)$ that satisfy the CLOPA constraint in (13). We then proceed with the SIS design to produce an architecture \mathcal{A} . The architecture is then verified to estimate its probability of failure on demand or equivalently its RRF_v . The architecture is also used to carry out a security risk assessment to estimate the probability of security failures $P[A'_S]$ and $P[A'_{BS}]$. If the new set of obtained values $(P[A'_S], P[A'_{BS}], RRF_v)$ still satisfy the CLOPA equation, the design stops. Otherwise, a new iteration will start to adjust the design in order to achieve the CLOPA constraint. This adjustment could be either by adding more security controls or by increasing the reliability of the

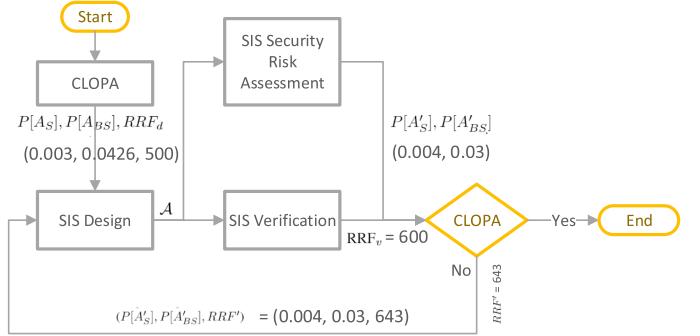


Fig. 4. CLOPA iterative design process. CSTR case study design values are shown.

system using fault-tolerant techniques. Algorithm 1 summarizes the iterative design process.

One question is how can we choose the initial values for RRF , $P[A_S]$, $P[A_{BS}]$? This initial design point could be selected with the aid of the design contour plot as in Fig. 3, where the design point strikes a balance between security and reliability. What is *reasonable* regarding the RRF is well documented in the standards using SILs, and the extra cost to move from one SIL to a higher SIL is well quantified in industry. What is not very clear, though, is what is an achievable value for the probability of security failures. This is still not a well-developed field, and the argument of how to assess such probabilities is still going on in the research community.

Another important question is whether there is any formal guarantees that Algorithm 1 will terminate. To answer this question, we need to know, or at least approximate, how the architectural design \mathcal{A} impacts the RRF, $P[A_S]$, and $P[A_{BS}]$. As pointed out earlier, this is very hard in practice. Without such relationship, the question of convergence to a solution for algorithm termination cannot be precisely answered. However, in practice, modifying the SIS design to increase the RRF is usually done by changing sensor and actuator configuration or reliability figures, as they are often the weakest links in the reliability chain, while the logic solver is minimally changed [4]. Accordingly, for all practical purposes, we can assume that the design process will converge after few runs.

B. Integrated Safety-Security Lifecycle

As the analysis in this article shows a clear coupling between safety and security design requirements, we propose the integrated lifecycle in Fig. 5. In the following, we present a brief description of the lifecycle steps in the order of their execution, according to the numbering labels in Fig. 5.

- ① *SIS safety lifecycle—HAZOP*: The first step is to carry out the hazard analysis for the physical system, often using HAZOP. This process identifies important assets that may be subject to, or contribute to, risk scenarios. Then, the process identifies all feasible hazards and associated risk ranking, as well as the associated cyber components for each identified hazard. This constitutes an input to the BPCS security lifecycle. If we designate the set of

Algorithm 1: Integrated Safety-Security Lifecycle Design Algorithm.

```

input : BPCS
output:  $\mathcal{A}, \theta_S$ 
 $([P[A_B], P[A_{SB}]) \leftarrow \text{BPCS-SecCycle(BPCS)};$ 
 $\theta_B \leftarrow (P[A_B], P[A_{SB}]);$ 
 $(P[A_S], P[A_{BS}], \text{RRF}_d) \leftarrow \text{DesignContour}(\theta_B);$ 
 $\theta_S \leftarrow (P[A_S], P[A_{BS}], \text{RRF}_d);$ 
do
     $\mathcal{A} \leftarrow \text{SIS-SafeCycle}(\theta_S);$ 
     $\text{RRF}_v \leftarrow \text{SIS-Verify}(\mathcal{A});$ 
     $(P[A'_S], P[A'_{BS}]) \leftarrow \text{SIS-SecCycle}(\mathcal{A});$ 
     $\text{RRF}' \leftarrow \text{CLOPA}(P[A'_S], P[A'_{BS}], \theta_B);$ 
     $\theta_S \leftarrow (P[A'_S], P[A'_{BS}], \text{RRF}');$ 
while  $\text{RRF}_v < \text{RRF}'$ ;
return  $\mathcal{A}, \theta_S$ ;

```

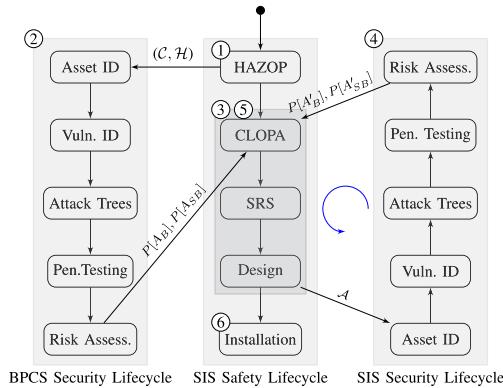


Fig. 5. Integrated safety and security lifecycles. The process starts at ① HAZOP, followed by ② BPCS complete security lifecycle, then ③ SIS safety lifecycle up to the end of the design stage, followed by ④ complete SIS security lifecycle, ⑤ CLOPA check, and possibly several iterations of steps ③, ④, and ⑤ and then terminates at the SIS installation stage.

hazards by \mathcal{H} , and the set of cyber components by \mathcal{C} , then the output from this process is the function $f : \mathcal{H} \mapsto \mathbb{R}$ representing the risk ranking and the relation $R \subseteq \mathcal{H} \times \mathcal{C}$ representing the cyber components for each hazard.

- ② **BPCS cyber security lifecycle:** The BPCS cyber security lifecycle, including vulnerability analysis, attack tree generation, penetration testing, and risk assessment, is performed on the BPCS. Ideally, the security risk assessment should be carried out for each process hazard scenario identified during HAZOP to identify the relevant vulnerabilities that may cause a process disruption. However, for the given centralized architecture, the BPCS is typically controlling a large number of control loops; hence, it may not be necessary to repeat the security risk assessment process for each control loop, as vulnerabilities may be applicable to several hazardous scenarios. The output of this process is the BPCS security failure probabilities $P[A_B]$ and $P[A_{SB}]$.
- ③ **SIS safety lifecycle—CLOPA and SIS design:** The first iteration of CLOPA and SIS design will proceed

according to Algorithm 1 and Fig. 4. The CLOPA calculates the design requirement for the SIS in terms of its reliability as defined by the RRF, and its cyber security resilience as defined by $P[A_S]$ and $P[A_{BS}]$. The SIS design then proceeds according to IEC 61511 standard [4] to produce an architecture \mathcal{A} . The design includes the hardware architecture, redundancy scheme, and software architecture. The specific design architecture can vary across industries and organizations, but the design has to achieve the required RRF, $P[A_S]$ and $P[A_{BS}]$, as calculated by CLOPA. After the design is completed, SIS verification is carried out to calculate the RRF_v . It should be highlighted that the SIS is one component only of the SIF. The SIF includes the sensor, SIS, and the actuator. Therefore, the verification is carried out on the whole SIF. For a detailed discussion on SIS design and verification, the reader is referred to [3].

- ④ **SIS cyber security lifecycle:** Using the resulting SIS design hardware and software architecture \mathcal{A} , the SIS security lifecycle is carried out. Since the SIS is not yet implemented at this stage, SIS penetration testing is not possible and, hence, omitted from the security lifecycle. The output from this process is the SIS security failure probabilities $P[A'_S]$, $P[A'_{BS}]$, derived from SIS vulnerabilities that may lead to a process hazard. It is noted that the SIS security lifecycle at the right of Fig. 5 proceeds from bottom to top for a better presentation.
- ⑤ **Safety lifecycle—CLOPA:** The CLOPA calculation is carried out using the values obtained from the safety verification and SIS security lifecycle, $(P[A'_S], P[A'_{BS}], \text{RRF}_v)$, to verify that the architecture \mathcal{A} satisfies the CLOPA constraint. The process SIS safety lifecycle → SIS Cyber security lifecycle → CLOPA (designated by the blue arrowed arc in Fig. 5) repeats until the CLOPA constraint is satisfied.
- ⑥ **Installation:** The finalized design then moves to the installation phase.

V. INTEGRATED DESIGN EXAMPLE

In this section, we present an integrated design example for a process control system to illustrate the proposed CLOPA and integrated lifecycle. The system described in this section is a real testbed located in Qatar University and comprises the process simulator and the full plant control system. As the integrated design lifecycle is substantial, with some steps outside the scope of this article (e.g., SIS architectural design and security risk assessment), it is not possible to present the design process in full details. However, we try to focus on the big picture as related to the proposed CLOPA, while discussing briefly each design step. Wherever needed, we refer the reader to relevant references for further details.

A. CPS Description

We consider the continuous stirred tank reactor (CSTR) process illustrated in Fig. 6. The reactor vessel has an inlet stream carrying the reactant A, an outlet stream carrying the product

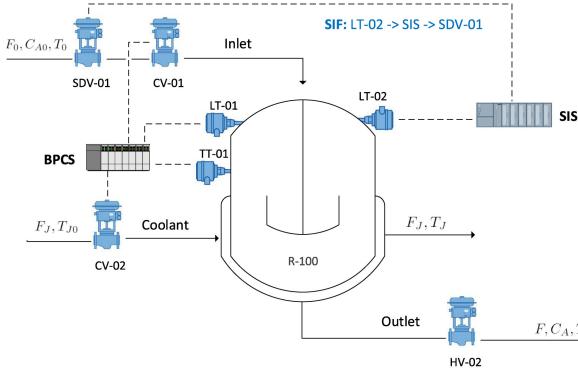


Fig. 6. Reactor piping and instrumentation diagram. ISA standard symbols are not strictly followed for illustration purposes.

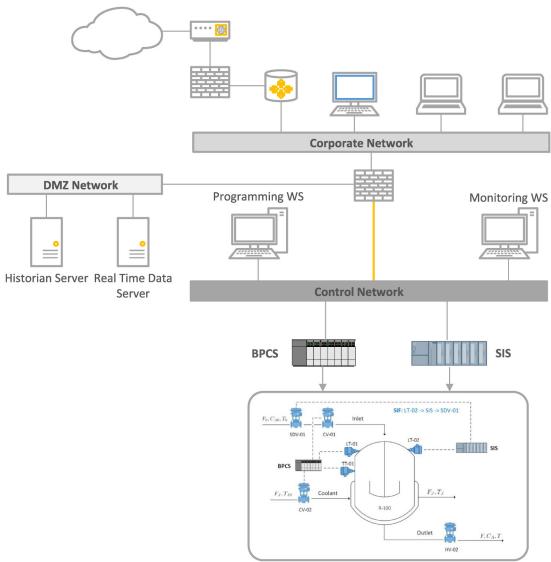


Fig. 7. CPS architecture for an industrial control system testbed, following NIST 800-82 guidelines. The firewall blocks any direct communication between the corporate network and the control network. Plant floor information is accessible only via the DMZ.

B, and a cooling stream carrying the cooling fluid into the surrounding jacket to absorb the heat of the exothermic reaction. A first-order reaction takes place where a mole fraction of reactant A is consumed to produce product B. The process has a level control loop ($LT-01 \rightarrow BPCS \rightarrow CV-01$) to maintain the liquid level in the reactor, and a temperature control loop ($TT-01 \rightarrow BPCS \rightarrow CV-02$) to control the reaction rate. The SIF ($LT-02 \rightarrow SIS \rightarrow SDV-01$) protects the reactor from the overflow hazard and will be explained later in this section. For more detailed explanation about the process including the state space model, the reader is referred to [12].

The CSTR process is controlled by the industrial control system shown in Fig. 7, which follows NIST 800-82 standard with one firewall and a DeMilitarized (DMZ) zone [13]. The corporate network cannot communicate directly with the control network. The only allowable information flow paths via the firewall are from the control network to the data logging servers in the DMZ zone and from the corporate network to the DMZ for

information retrieval. The BPCS and the SIS have Modbus/TCP communication over the control network [6].

B. Integrated Lifecycle

In the following discussion, we follow the integrated lifecycle in Fig. 5, and as per the itemized steps in Section IV-B, we have the following.

(1) *SIS safety lifecycle—HAZOP*: Table II shows the HAZOP sheet for the CSTR process. Each row contains: 1) the possible hazard; 2) all possible initiating events for each hazard whether mechanical or electronic failures; 3) consequences if the hazard occurred, including safety, financial, and environmental losses; 4) existing safeguards that could prevent the hazard from propagating and causing the consequences; and 5) the risk rank, which is typically a function of the consequences. There are two identified hazards for the reactor process: high level causing an overflow hazard, and high temperature that may lead to reactor runaway and possible meltdown. Both the hazards have high and very high risk rankings; therefore, the two risk scenarios qualify for further LOPA assessment. In the following, we limit our discussion to the high-level hazard scenario only. High-temperature hazard could be treated similarly.

(2) *BPCS cyber security lifecycle*: We need to calculate $P[A_B]$ and $P[A_{SB}]$ for the BPCS, the probability that the BPCS fails due to a direct attack and a SIS-pivot attack, respectively, in a way that generates the high-level process hazard. We conducted vulnerability identification on the CPS network in Fig. 7, constructed the attack trees, and carried out penetration testing to verify the vulnerability findings. We assumed an attacker profile where attacks are selected randomly. The total number of semantically relevant attacks is found to be $|\mathcal{A}_r| = 42$. We assume that relevant attacks represent 10^{-4} of all possible attacks, i.e., $|\mathcal{A}_r|/|\mathcal{A}| = 10^{-4}$. Therefore, according to Lemma 3.1, we obtain an initiating event likelihood $10^{-4}\lambda$ and a weight factor $\gamma_a = 0.024$ for each attack a at the leaf nodes of the attack tree. As the full details of vulnerability analysis, attack design, and penetration testing are beyond the scope of this article, we refer the interested reader to [10] and [14].

To compromise the BPCS, we assume the more realistic situation with no insider threat and no direct communication from the corporate network to the control network. In this scenario, the attacker has to detour to compromise the real-time (RT) server in the DMZ and use it as a pivot to attack the BPCS, either directly or via the monitoring workstation (designated HMI hereafter) that has legitimate communication with the BPCS. We start with the assumption that one of the corporate network PCs that has legitimate access to the RT server is compromised. There are several well-known attack vectors in the IT security domain to achieve such compromise, such as a spam email, a web service vulnerability, or an external malware USB, just to name a few. Fig. 8 shows an abstract attack tree that summarizes the BPCS compromise paths, where the database server compromise is a prerequisite attack step. In the following, we expand each of the leaf nodes in this abstract attack tree into the corresponding detailed attack trees. More detailed treatment of each attack tree as well as penetration testing could be found in [10].

TABLE II
PARTIAL HAZOP SHEET FOR THE REACTOR PROCESS

Hazard	Initiating Event (Cause)	Consequences	Safeguards (IPL)	Risk Rank
High Level (Reactor overflow)	BPCS failure OR Human error (mis-aligned valves)	2 or more fatalities (safety), Product loss (financial), Environmental contamination (environment)	Reactor dike (Mitigation)	High
High Temperature (Reactor Meltdown/explosion)	BPCS failure OR Coolant inlet control valve fully (partially) closed OR Inlet valve stuck fully open	10 or more fatalities (safety), Product loss (financial), Environmental contamination (environment)	None	V. High

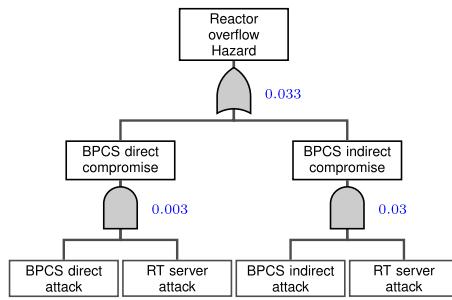


Fig. 8. Abstract attack tree to compromise the BPCS to generate overflow process hazard for the CSTR reactor. Leaf nodes are further expanded in Figs. 9–11.

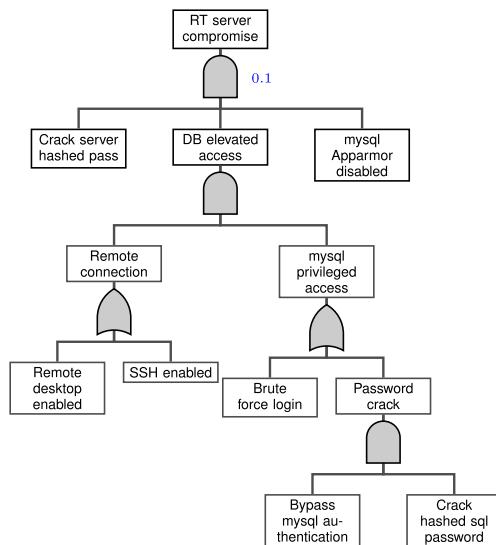


Fig. 9. RT server attack tree. Database vulnerabilities are exploited to gain root access and use the RT server as a pivot to attack the control network. DB elevated access is a prerequisite to crack the server hashed password. Leaf nodes that represent a distinct attack have a weight factor $\gamma_a = 0.024$ that is combined with their success probability. Therefore, Bypass mysql and Crack hashed sql password, Brute force login, SSH enabled, remote desktop enabled, and each has a weight factor combined with their success probability.

Fig. 9 shows the RT server attack tree. The basic idea is to exploit mysql database vulnerabilities via SSH to obtain the Linux server password Hash Dump and possibly crack the password to achieve privilege escalation and gain full control over the RT server. The probability of success of such an attack depends on several factors, including mysql configuration settings to allow brute-force login attack, mysql login password strength,

configuration of mysql security monitoring app, and whether SSH is enabled. For the purpose of this case study, we choose this probability arbitrarily as 0.1. It should be highlighted that the attack tree does not have the sequence semantics to represent a sequence of attack steps. For example, the remote connection step has to be executed before the mysql privileged access in Fig. 9. We represent this sequence by the AND gate aggregator, noting that in some other cases the AND gate may represent simultaneous attack steps. For more information on attack trees and their semantics, the reader is referred to [15].

Fig. 10 shows the BPCS attack tree, which is divided into two main parts: DoS attack and integrity attack. The DoS attack may not lead to a reactor overflow unless there is a concurrent process disturbance that could not be controlled with the DoS-induced delayed BPCS control response. The probability of such disturbance could be estimated from plant information. The integrity attack injects a low-level measurement value for LT-01 to drive the BPCS controller to increase valve CV-01 opening or directly forces control valve CV-01 to open 100%. This will cause a reactor overflow if the SIS is not activated. The injection of the malicious value in the control loop could be accomplished by either gaining access to the controller and overwriting the control program or more simply sending Modbus packets to the controller with the malicious values. Modbus attack is much easier to launch but requires configuration data to identify the Modbus register address for either LT-01 or CV-01. The probability of BPCS indirect attack is chosen arbitrarily as 0.03.

Finally, Fig. 11 shows the attack tree for BPCS attack via the HMI. The attack is launched by remote desktop connection to the HMI and legitimately controlling CV-01 via the GUI. This indirect attack is easier than targeting the BPCS directly as it does not require knowledge about the controller configuration or Modbus register addresses associated with the sensor and valve of the targeted control loop. This is because all the information is already programmed in the GUI software. The probability of BPCS indirect attack is estimated to be 0.3. Using Fig. 8 and the three presented attack trees in Figs. 9–11, the total probability of BPCS attack that leads to an overflow hazard could be estimated by $P[A_B] \approx 0.033$.

It should be highlighted that the assignment of a probability measure to the success of attack actions is subject to debate in the research community, and there is no published agreed-upon data as in the case of reliability failure data. One approach is to use attack databases, such as NIST National Vulnerability Database (NVD) [16], to estimate the probability of a cyber attack success based on attributes such as required knowledge level and attack

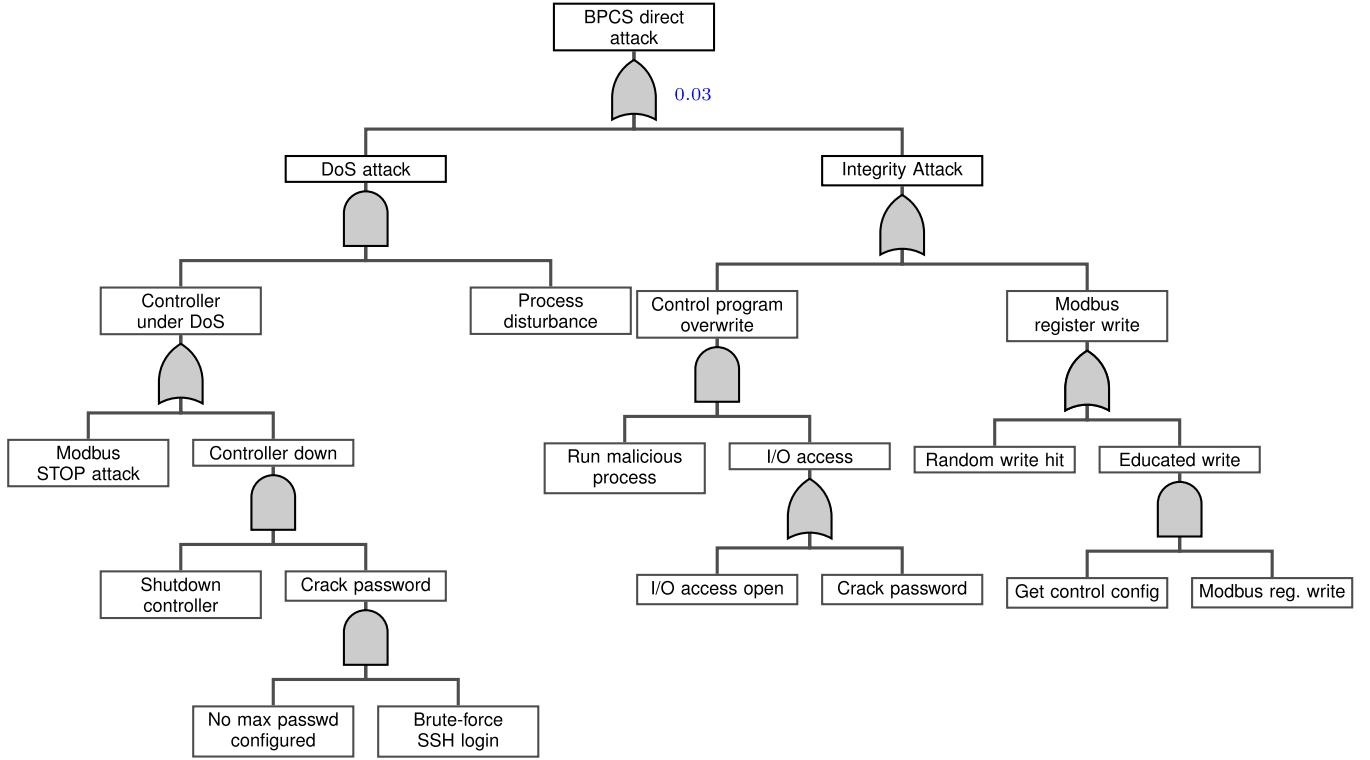


Fig. 10. Attack tree for BPCS compromise to generate a reactor overflow hazard. A DoS attack synchronized with a process disturbance or an especially crafted integrity attack would cause the CSTR to overflow. Leaf nodes that represent a distinct attack have a weight factor $\gamma_a = 0.024$ that is combined with their success probability.

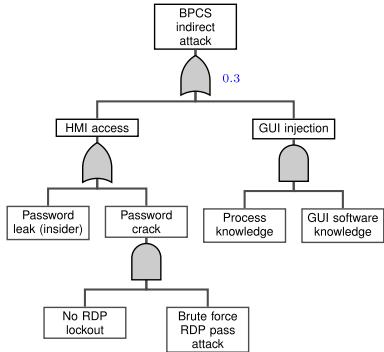


Fig. 11. HMI-BPCS indirect attack tree. The compromised HMI is used to embed the attack against the BPCS using the legitimate traffic between the GUI and the BPCS control program. Leaf nodes that represent a distinct attack have a weight factor $\gamma_a = 0.024$ that is combined with their success probability.

difficulty. However, this approach has the drawback that it does not take into account the specifics of each organization. In this article, we rely on the experience obtained during the penetration testing carried out by the research team in combination with NVD to assign the probability measures. This does not impact the analysis as the presented case study is meant for illustration purposes to explain the design process.

To calculate the probability of BPCS cyber attack leading to a process hazard given an SIS cyber compromise $P[A_{SB}]$, we focus on Modbus attack vectors for both integrity and DoS attacks. Integrity attacks target sensor LT-01 or valve CV-01

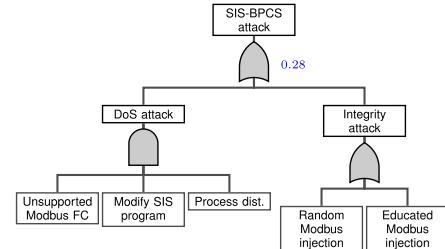


Fig. 12. SIS-BPCS attack tree. The SIS is used as a pivot to launch either a DoS attack or a crafted integrity attack that results in reactor overflow. Leaf nodes that represent a distinct attack have a weight factor $\gamma_a = 0.024$ that is combined with their success probability.

as before, either randomly or using leaked Modbus register configuration. DoS attack could be launched by utilizing non-programmed Modbus function code hoping that it would crash the BPCS Modbus master. Fig. 12 summarizes the attack tree, and the probability is chosen arbitrarily as $P[A_{SB}] \approx 0.2813$. To summarize, the desired outcome from the BPCS security lifecycle is $(P[A_B], P[A_{SB}]) = (0.033, 0.2813)$.

It should be noted that complete attack trees for the given BPCS and CPS architecture could span multiple pages. However, full attack trees may obscure the analysis and will serve no additional insight. Therefore, the simplified attack trees presented here act as a better illustration of the design methodology. For more in-depth treatment of the cyber risk assessment for the presented case study, refer to [10].

TABLE III
LOPA SHEET FOR THE CSTR OVERFLOW HAZARDOUS SCENARIO

Initiating Event	Likelihood λ_i (/yr)	Tank Dike	Safety Procedure	Human Intervention	BPCS ($P[B_p]$)	TMEL
Inlet flow surge	10^{-1}	10^{-2}	1	10^{-1}	10^{-1}	10^{-6}
Downstream flow blockage	10^{-1}	10^{-2}	10^{-1}	10^{-1}	10^{-1}	10^{-6}
Manual valves misalignment	10^{-1}	10^{-2}	10^{-1}	10^{-1}	10^{-1}	10^{-6}
BPCS physical Failure	$10^{-1}(\lambda_b)$	10^{-2}	1	10^{-1}	1	10^{-6}
BPCS attack Failure	$10^{-2}(\lambda_c)$	10^{-2}	1	10^{-1}	1	10^{-6}

Numbers in each cell represent the probability of failure of the associated protection layer.

TABLE IV
CSTR CLOPA—CALCULATED PARAMETER VALUES

LOPA Parameter	Value	Source
$\sum_{i=1}^3 \lambda_i$	0.3	LOPA Sheet
$P[\bar{L}]$	0.001	LOPA Sheet
λ_b	0.01	LOPA Sheet
λ_c	0.01	LOPA Sheet
$P[B_p]$	0.01	LOPA Sheet
α_1	1.13×10^{-4}	CLOPA Parameter-Calculated Eq. (10)
α_2	1.17×10^{-4}	CLOPA Parameter-Calculated Eq. (11)
β	10^{-6}	CLOPA Parameter-Calculated Eq. (12)
γ_1	1.4868×10^{-4}	CLOPA Parameter-Calculated Eq. (14)
γ_2	7.5785×10^{-6}	CLOPA Parameter-Calculated Eq. (15)
γ_3	1.1686×10^{-4}	CLOPA Parameter-Calculated Eq. (16)

③ *SIS safety lifecycle—CLOPA:* Table III shows the LOPA sheet for the CSTR overflow hazard identified from the HAZOP, where the initiating event likelihoods and failure probabilities are adopted from [17] and [18]. The BPCS cyber attack likelihood is calculated as $\lambda_c = 10^{-4}\lambda = 0.01$ per year, assuming $\lambda = 100$ per year. Note that human intervention is considered a protection layer assuming that there is sufficient time for the operation team to manually isolate the reactor in the field. Some conservative approaches omit any human intervention or safety procedure from the LOPA.

From the LOPA sheet, we extract the event likelihood values to calculate the CLOPA model parameters using (10)–(12) and (14)–(16), along with $(P[A_B], P[A_{BS}]) = (0.033, 0.2813)$ from the BPCS security lifecycle. Table IV summarizes the parameter values. Substituting into the CLOPA constraint (13), we obtain

$$P[S_p] \leq \frac{1 - 148.68P[A_S] - 7.6P[A_{BS}](1 - P[A_S])}{117(1 - P[A_S]) - 7.6P[A_{BS}](1 - P[A_S])}. \quad (27)$$

Our objective now is to design an SIF with architecture \mathcal{A} that satisfies (27) in order to achieve the required process safety objective as defined by the TMEL in the LOPA analysis. Our initial design for the SIF will comprise a level sensor (LT-02), a logic solver (SIS), and a shutdown valve (SDV-01), as illustrated in Fig. 6. The SIF will take an independent action upon reactor overflow and will close the inlet shutdown valve. The architecture of the SIF could vary through design iterations to achieve the required safety. As an example, sensors may be duplicated or sometimes triplicated to achieve higher reliability, and the SIS architecture may include redundant CPU modules. We note that for a perfectly secured SIS ($P[A_S] = P[A_{BS}] = 0$), $P[S_p] \leq 1/117$, or equivalently $RRF \geq 117$. This is the minimum achievable RRF. Since for practical systems there is no

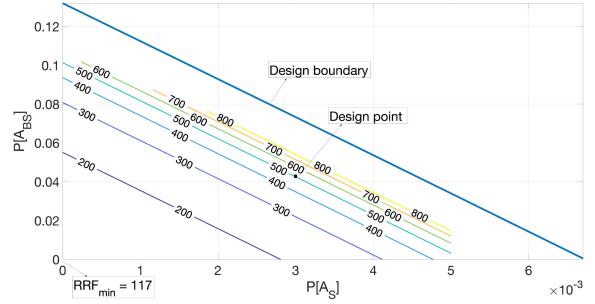


Fig. 13. CSTR case study: CLOPA design region with the contour plot for the RRF.

zero probability of cyber security attack failures, our SIS design is expected to have an $RRF > 117$.

Using the calculated LOPA parameter values, the design region (18) and boundary (19) are defined by

$$P[A_{BS}] \leq 0.132 \left(\frac{1 - 148.68P[A_S]}{1 - P[A_S]} \right) \quad (28)$$

where the design boundary is defined when the equality holds. The contour lines for the RRF in (20) are defined by

$$P[A_{BS}] = \left(\frac{15.42}{C - 1} \right) \left(\frac{(C - 0.008) - (C - 1.27)P[A_S]}{1 - P[A_S]} \right) \quad (29)$$

for different values C of the RRF. The design region and the contour lines are plotted in Fig. 13. We note that as we approach the design boundary, either by increasing $P[A_S]$ or $P[A_{BS}]$, the RRF rapidly increases such that it is not possible to plot the contour lines in this region in a visible way. The design in this region is very sensitive to input variations (i.e., a very small variation in probabilities will result in a very large change in RRF). Therefore, the design point should be selected as far as possible from the design boundary. To further illustrate the increase in RRF, Fig. 14 shows a 3-D plot for the RRF as it varies with both $P[A_S]$ and $P[A_{BS}]$. It should be evident from the 3-D plot that for small values of $P[A_S]$, the function gradient is smaller, resulting in a less-sensitive design to probability variations. At larger values of $P[A_S]$ near the design boundary, the RRF increases exponentially with $P[A_{BS}]$. These results could be verified by calculating the gradient of (13).

To proceed with the design process, we pick the point $P[A_S] = 0.003$ as a reasonable probability value for SIS direct attack failure that is away from the steepest ascent region in Fig. 14. We now need to choose a practical value of $P[A_{BS}]$

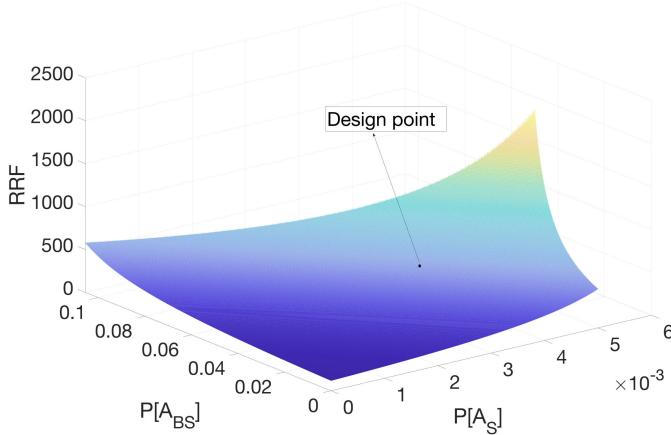


Fig. 14. CLOPA RRF as it varies with SIS security failure probabilities. Steepest ascent region to the right should be avoided when selecting the operating point.

that results in an achievable target RRF. With the help of Fig. 13 and contour lines, $P[A_S] = 0.003$ intersects the contour line for $\text{RRF} = 500$ at $P[A_{BS}] = 0.0426$. Alternatively, the value of $P[A_{BS}]$ could be obtained from (29) by setting $C = 500$ and $P[A_S] = 0.003$. The design point $(0.003, 0.0426, 500)$ is indicated in Figs. 13 and 14. The design and verification of the SIF then resume according to IEC 61511 to develop an architecture \mathcal{A} that satisfies the combined CLOPA requirement: $\text{RRF} \geq 500$, $P[A_S] \leq 0.003$, and $P[A_{BS}] \leq 0.0426$. The detailed design and verification of the SIF are outside the scope of this article (refer to [4] for more details). To complete the case study, we will assume that the design engineer came up with an architecture \mathcal{A} that was verified using vendor data, resulting in reliability $\text{RRF}_v = 600$, with a design margin from the required $\text{RRF} = 500$.

④ *SIS cyber security lifecycle*: The resulting SIF architecture \mathcal{A} is used to carry out the SIS security lifecycle, similar to the BPCS security risk assessment in step 2 of the design process. As the SIS detailed design and verification are not in the scope of this article, we will assume for the sake of illustration that the architecture \mathcal{A} results in a cyber system configuration that has a higher probability of SIS security attack failure $P'[A_S] = 0.004$ while reducing the BPCS pivot attack failure probability to $P'[A_{BS}] = 0.03$ via securing the BPCS–SIS link.

⑤ *Safety lifecycle—CLOPA*: The architecture \mathcal{A} results in $P'[A_S] = 0.004$, $P'[A_{BS}] = 0.03$, and $\text{RRF}_v = 600$. We need to verify if these values satisfy the CLOPA constraint (27). Plugging the probability values results in $P[S_p] \leq 1.54 \times 10^{-3}$, or equivalently $\text{RRF} \geq 643$. As $\text{RRF}_v = 600 < 643$, the architecture has to be modified either by reducing further the cyber attack failure probabilities or by increasing the system reliability via fault tolerance techniques. It may take the design engineer multiple iterations until the design achieves the CLOPA constraint. In practice, the iterations do not involve a complete architectural redesign, but rather changing the redundancy scheme or security hardening in order to achieve the design objective. To conclude the case study example, we will assume that the design engineer came up with an architecture that preserves the aforementioned probability values while increasing the RRF to

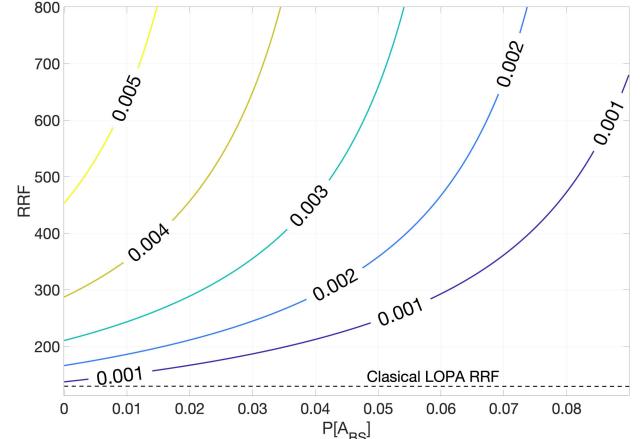


Fig. 15. Increase of the RRF with $P[A_{BS}]$. Each curve corresponds to the fixed value indicated for $P[A_S]$.

650. This concludes the design process and the system moves to the implementation phase. The case study design values are superimposed on the iterative design process in Fig. 4 as an illustration. ■

C. Classical LOPA Error

Classical LOPA ignores cyber attack probabilities altogether. For the given problem, it results in $\text{RRF} = 113$ as per (17). The minimum CLOPA RRF occurs for a perfectly secured safety system where $P[A_S] = P[A_{BS}] = 0$, achieving $\text{RRF} = 117$. Therefore, the minimum error between LOPA and CLOPA RRF estimation is 4. The error gets worse as security failure probabilities increase. For the given design point $P[A_S], P[A_{BS}] = (0.003, 0.0426)$, the classical LOPA error is $e_{\text{RRF}} = 378$. This is a significant amount of error that results in the design of a less reliable system that will not achieve the target risk level. Fig. 15 better illustrates the error increase with increasing the security failure probability $P[A_{BS}]$ for different values of $P[A_S]$. For small values of $P[A_S]$, the curves show slow increase in the RRF with $P[A_{BS}]$. As $P[A_S]$ increases, the RRF increase becomes exponential. A similar contour figure for fixed $P[A_{BS}]$ values could be generated. The design point for the case study $P[A_S] = 0.003$ was chosen as a tradeoff between an achievable cyber attack probability value and a moderate rate of increase for the RRF. The 3-D plot for the error in RRF versus $P[A_S], P[A_{BS}]$ is identical to Fig. 14, except by shifting down the 3-D curve by 113, the LOPA RRF value; therefore, it is omitted to avoid repetition.

D. Sensitivity Analysis

Calculating the probability of a security failure is a debatable subject in the research community, especially with the lack of statistical data that are available for physical failures. One question that comes to mind is the robustness of the developed CLOPA model to probability variations. We conducted a numerical analysis to calculate the partial derivatives of the RRF with respect to $P[A_S], P[A_{BS}]$. The two partial derivative plots are very similar to Fig. 14 and omitted for space limitation. For

small probability values, the change in the RRF is in the range of 15% for 10^{-3} change in $P[A_S]$. As probabilities increase and we approach the decision boundary, the change in the RRF jumps to around 80% for 10^{-3} change and increases exponentially as we get closer to the decision boundary. A similar behavior is exhibited with $P[A_{BS}]$ change (figure omitted for brevity). However, the change in the RRF has much lower percentage, ranging from 7% for small probability values, and increasing to around 37% as we approach the decision boundary. We highlight the following three key observations.

- 1) For small cyber failure probability values, the model sensitivity is acceptable since the SIL levels have an order of magnitude ratio, so a small percentage change would likely keep the system requirement in the same SIL category. However, this requires that the probability error is in the range of 10^{-3} .
- 2) The model is more sensitive to direct attack failure probabilities than BPCS pivot attacks.
- 3) We should always try to design our system as far as possible from the decision boundary. The model sensitivity with respect to probability changes increases as we approach the decision boundary.

VI. RELATED WORK

HAZOP has been the dominant risk assessment method for the process industry for over 30 years [5], [19], [20]. LOPA has been used in conjunction with HAZOP to design SISs and specify the SIL for each SIF [21]. Because of the wide adoption of LOPA by industry due to its systematic approach and quantitative risk assessment capability, LOPA has been included as one of the methods in IEC 61511-3 standard with several illustrating examples [4]. The LOPA approach has been applied to physical security risk analysis in [22]. However, to the best of authors' knowledge, there is no research work on integrating security attacks in the LOPA framework for SIS design.

There are emergent standardization initiatives to address safety and security coordination in CPSs. IEC 62443-4-1 (Security for Industrial Automation and Control Systems—Part 4-1: Secure Product Development Lifecycle Requirements) is a standard developed by the ISA-99 committee with the purpose to extend the existing safety lifecycle at different phases to include security aspects to ensure safe CPS design [23]. IEC TC65 AHG1 is a recently formed group linked to the same technical committee developing IEC 61508 and IEC 62443 to consider how to bridge functional safety and cyber security for industrial automation systems [24]. IEC 62859 (Nuclear Power Plants—Instrumentation and Control Systems—Requirements for Coordinating Safety and Cyber Security) is a standard derived from IEC 62645 for the nuclear power industry to coordinate the design and operation efforts with respect to safety and cyber security [25]. DO-326 (Airworthiness Security Process Specification) is a standard for the avionics industry that augments existing guidelines for aircraft certification to include the threat of intentional unauthorized electronic interaction to aircraft safety [26]. A taxonomy of dependable and secure computing is introduced in [27] in order to facilitate the communication among different research communities. The concepts

and taxonomy presented are a result of a joint committee on “Fundamental Concepts and Terminology” that was formed by the Technical Committee on Fault-Tolerant Computing of the IEEE Computer Society and the IFIP WG 10.4 “Dependable Computing and Fault Tolerance.” A preliminary work on the research in this article that combines the two research directions stated below is presented in [28].

A. Lifecycle Integration

Kornecki and Liu [29] use fault tree analysis to combine both safety and security failures in one unified risk assessment framework for the aviation industry. The outcome of the risk assessment is used to define both safety and security requirements. A road map for cyber safety engineering to increase air traffic management system resilience against cyber attacks is proposed in [30]. The V-shaped model to develop embedded software for CPS is augmented with security actions in [31]. The integration of IEC 61508 safety standard and IEC 15408 for IT security is described in [32]–[34] for building automation systems. Sørby [35] describes in more detail the integration of IEC 61508 safety lifecycle and the CORAS approach to identify security risks [36]. An approach to align safety and security during the different stages of system development lifecycle is proposed in [37]. The approach, called Lifecycle Attribute Alignment, ensures compatibility between safety and security controls developed and maintained during the system development lifecycle. HAZOP, a predominantly used method for safety risk assessment in the process industry, is modified in [38] to include security failures. The authors introduce new guide words, attributes, and modifiers for security components akin to traditional HAZOP limited to safety failures. Failure Mode and Effect Analysis is extended in [39] to include security vulnerabilities, suggesting the name Failure Mode Vulnerability and Effect Analysis. For a survey on the integration of safety and security in the CPS, refer to [2].

B. Model-Based Risk Assessment

Several graphical methods have been used to combine safety and security analysis. Goal structuring notation (GSN) is a graphical notation used to model requirements, goals, claims, and evidence of safety arguments [40]. The SafSec research project for the avionics industry elaborates on the use of GSN to integrate both safety and security arguments in one representation [41]. A similar approach is used in [42], where the authors apply the nonfunctional requirement (NFR) approach to quantitatively assess the safety and security properties of an oil pipeline CPS. NFR is a technique that allows simultaneous safety and security graphical representation and evaluation at the architectural level.

The simplicity and wide adoption of fault and attack trees promoted the research work to merge both modeling tools. The integration of fault trees and attack trees is considered in [43] in order to extend traditional risk analysis to include cyber attack risks. A quantitative analysis is proposed by assigning probabilities to tree events. Similarly, fault tree analysis is used in [29] to analyze safety/security risks in aviation software. Steiner and Liggesmeyer [44] extend component fault trees to

contain both safety and security events. Both the qualitative and quantitative analyses are performed to assess the overall risk. The quantitative analysis is enabled by assigning probabilities to safety events and categorical rating (low, medium, and high) for security events. Kumar and Stoelinga [45] translate the combined fault-attack tree into stochastic time automata to enable quantitative risk analysis. The use of bow-tie diagrams and analysis in place of fault trees is reported in [46], where it is integrated with attack trees for combined safety–security risk assessment.

Given the limited semantics of fault trees, Boolean-logic-driven Markov process (BDMP) graphical formalism introduced in [47] has been used to integrate safety and security events. The approach integrates fault trees with the Markov process at the leaf node level and associates a mean time to success for security events and a mean time to failure for safety events. This allows both a qualitative and a quantitative risk assessment for the given system. The formalism also enables the modeling of detection and response mechanisms without a need for model change. The work in [48] applies BDMP formalism to a pipeline case study, illustrating different types of safety–security interdependencies. In [49], Stuxnet attack is modeled using the BDMP and a quantitative risk analysis is carried out on the industrial control system.

Petri nets have also been proposed to overcome the limitations of fault trees. A formalism for safety analysis named state/event fault trees is reported in [50]. In this formalism, both deterministic state machines and Markov chains are combined, while keeping the visualization of causal chains known from fault trees. This formalism is extended in [51] to include an attacker model to deal with both safety and security. Similarly, stochastic Petri nets have been used in [52] to model the impact of intrusion detection and response on CPS reliability and in [53] to assess the vulnerabilities in supervisory control and data acquisition systems. Bayesian belief networks are also considered as one of the model-based approaches. In [54], a Bayesian belief network is used to assess the combined safety and security risk for an oil pipeline example.

The Unified Modeling Language (UML) commonly used in software engineering has also been used for safety and security risk assessment. Misuse cases for UML diagrams have been used to define safety requirements in [55] and security requirements in [56], independently. A combined process for Harm Assessment of Safety and Security has been proposed in [57] based on both UML and HAZOP studies. UMLsafe [58] and UMLsec [59] are two UML extensions that enable modeling of safety and security requirements, respectively. The combined UMLsafe/UMLsec is proposed in [60] for safety–security code-development. SysML-sec, a SysML-based model driven engineering environment, is used in [61] for the formal verification of safety and security properties.

System-theoretic process analysis (STPA) was developed as a new hazard analysis technique to evaluate the safety of a system [62]. Friedberg *et al.* [63] extend the STPA to include system security aspects in the analysis. The expanded approach is named STPA-SafeSec and demonstrated on a use case in the power grid domain. The system-theoretical accident model and process (STAMP) is applied to the Stuxnet attack in [64],

showing that the attack could have been avoided if the STAMP was applied during design time.

VII. CONCLUSION

Classical safety assessment methods do not take into account failures due to cyber attacks. In this article, we showed quantitatively that overlooking security failures could bias the risk assessment, resulting in underdesigned protective systems. In addition, the design of safety and security subsystems for complex engineering systems cannot be carried out independently, given their strong coupling as demonstrated in this article. Although the design becomes more complicated when considering cyber attacks, the development of new software tools or the modification of existing industrial tools could automate the process.

In this article, we considered the control system (BPCS) design as given, following common industrial practice. The joint optimization of both BPCS and SIS designs, from both safety and security perspectives, is a potential extension for the presented work. In addition, the presented integrated lifecycle relies in part on designer’s experience to make design decisions to achieve the system requirements. Optimal system design that captures possible safety and security design choices with associated financial cost could provide a better quantitative approach to find the optimal system operating point rather than relying on design heuristics. Furthermore, the integration of both the safety and security lifecycles into model-based design toolchains is crucial for adoption by industry.

Finally, the work presented in this article discusses the impact of cyber security failure on system safety. A closely related problem is how safety failures could impact cyber security. There is not much work in this direction, perhaps because the focus in CPSs is always on safety, considering the security of the cyber system as a secondary issue. Nevertheless, this is an important problem. On the one hand, a simple safety failure may be injected to cause a security compromise that may be exploited to produce a higher security compromise that could lead to a greater safety hazard. On the other hand, both directions, i.e., Safety → Security and Security → Safety, are closely related and interacting, and therefore, optimizing a CPS performance with respect to safety/security or both cannot be fully achieved without understanding the two types of interactions.

VIII. SOURCE CODE

The source code for the CLOPA in the form of MATLAB m files to regenerate the research results including the case study is located at <https://github.com/Ashraf-Tantawy/CLOPA.git>.

APPENDIX A

Proof of Lemma 3.1

Proof: The aggregate likelihood of all attacks to cause a hazard taking into account BPCS and SIS protection could be approximated by (neglecting higher order probability terms)

$$\Lambda = \lambda \sum_{a \in \mathcal{A}_r} \alpha_a P_a[S, B]. \quad (30)$$

TABLE A1
CLOPA MODEL PARAMETERS

Symbol	Description	Type	Calculation Method/ Data Source
λ_i	Initiating event i likelihood (/yr)	Parameter	Reliability data
λ_p	BPCS physical failure event likelihood (/yr)	Parameter	Reliability data
λ_a	BPCS cyber attack a likelihood (/yr)	Parameter	Refer to Section III-B
λ_c	BPCS semantically-related attacks likelihood (/yr)	Parameter	Refer to Section III-B, Lemma III.1
λ	BPCS cyber attack likelihood for all attacks	Parameter	Statistical attack data
α_a	Probability of selecting attack a by the attacker	Parameter	Attacker profile model
γ_a	Weight factor for the attacks for the equivalent BPCS	Parameter	Refer to Lemma III.1
$P[\mathcal{L}_i]$	Probability of failure of all protection layers for initiating event i	Parameter	Reliability data
TMEL	Target Mitigated Event Likelihood	Parameter	Determined by the corporate policy
$P[B_c]$	Probability of BPCS security failure	Intermediate design variable	BPCS security risk assessment
$P[B_p]$	Probability of BPCS physical failure	Parameter	Reliability data
$P[A_B]$	Probability of BPCS direct security failure	Parameter	BPCS security risk assessment
$P[A_{SB}]$	Probability of BPCS SIS-pivot security failure	Parameter	BPCS security risk assessment
$P[S_c]$	Probability of SIS security failure	Intermediate design variable	SIS security risk assessment
$P[S_p]$	Probability of SIS physical failure	Design variable	SIS security risk assessment
$P[A_S]$	Probability of SIS direct security failure	Design variable	SIS security risk assessment
$P[A_{BS}]$	Probability of SIS BPCS-pivot security failure	Design variable	SIS security risk assessment
$P[S_c, B_c]$	Probability of simultaneous SIS and BPCS security failure	Intermediate design variable	BPCS & SIS security risk assessment
$\alpha_1 - \alpha_2$	-	Auxiliary parameters	Eq. (10), (11)
$\gamma_1 - \gamma_3$	-	Auxiliary parameters	Eq. (14) to (16)
$\zeta_1 - \zeta_3$	-	Auxiliary parameters	Eq. (23) to (25)
β	-	Auxiliary parameters	Eq. (12)

Variables designated as “Design variable” are with respect to CLOPA, but could be a design variable of another assessment, such as $P[A_B]$, derived from BPCS security risk assessment. Variables designated as “intermediate design variables” could be expressed in terms of design variables.

Using (8) to expand the joint probability, we obtain

$$\Lambda = \lambda \sum_{a \in \mathcal{A}_r} \alpha_a (\eta_1 + \eta_2 P[B_c^a] + \eta_3 P[S_c] P[B_c^a | S_c]) \quad (31)$$

where $P[B_c^a]$ represents the probability of BPCS security failure with respect to attack a , and η_1 , η_2 , and η_3 are probability terms not dependent on the attack a . Expanding, we obtain

$$\Lambda = \lambda \left(\sum_{a \in \mathcal{A}_r} \alpha_a \right) \times \quad (32)$$

$$\left(\eta_1 + \eta_2 \sum_{a \in \mathcal{A}_r} \gamma_a P[B_c^a] + \eta_3 \sum_{a \in \mathcal{A}_r} \gamma_a P[S_c] P[B_c^a | S_c] \right) \quad (33)$$

where $\gamma_a = \alpha_a / \sum_{a \in \mathcal{A}_r} \alpha_a$. Ignoring higher order probabilities, we obtain

$$\Lambda \approx \lambda \left(\sum_{a \in \mathcal{A}_r} \alpha_a \right) \times \quad (34)$$

$$\left(\eta_1 + \eta_2 P \left[\sum_{a \in \mathcal{A}_r} \gamma_a B_c^a \right] + \eta_3 P \left[S_c, \sum_{a \in \mathcal{A}_r} \gamma_a B_c^a \right] \right). \quad (35)$$

Comparing (31) and (35), the second and third terms in (35) represent an equivalent BPCS with a combined attack vector \mathcal{A}_r , where each attack a is weighted by γ_a . In addition, the likelihood of this combined attack vector is $\lambda(\sum_{a \in \mathcal{A}_r} \alpha_a)$.

REFERENCES

- [1] S. Kriaa, L. Pietre-Cambacenes, M. Bouissou, and Y. Halgand, “A survey of approaches combining safety and security for industrial control systems,” *Rel. Eng. Syst. Saf.*, vol. 139, pp. 156–178, 2015.
- [2] X. Lyu, Y. Ding, and S. H. Yang, “Safety and security risk assessment in cyber-physical systems,” *IET Cyber-Phys. Syst.: Theory Appl.*, vol. 4, no. 3, pp. 221–232, 2019.
- [3] P. P. Gruhn, *Safety Instrumented Systems: Design, Analysis, and Justification*. Research Triangle Park, NC, USA: Instrum. Soc. Amer., 2006.
- [4] *Functional Safety—Safety Instrumented Systems for the Process Industry Sector—Part 1: Framework, Definitions, System, Hardware and Application Programming Requirements*, IEC Standard IEC 61511-1:2016, 2016.
- [5] J. Dunjó, V. Fthenakis, J. A. Vilchez, and J. Arnaldos, “Hazard and operability (HAZOP) analysis. A literature review,” *J. Hazardous Mater.*, vol. 173, no. 1–3, pp. 19–32, 2010.
- [6] A. Swales, *Open Modbus/TCP Specification*, vol. 29. Rueil-Malmaison, France: Schneider Electric, 1999.
- [7] I. N. Fovino, A. Carcano, T. De Lacheze Murel, A. Trombetta, and M. Maseri, “Modbus/DNP3 state-based intrusion detection system,” in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, 2010, pp. 729–736.
- [8] A. Z. Faza, S. Sedigh, and B. M. McMillin, “Reliability analysis for the advanced electric power grid: From cyber control and communication to physical manifestations of failure,” in *Proc. Int. Conf. Comput. Saf., Rel., Secur.*, 2009, pp. 257–269.
- [9] G. Stoneburner, A. Goguen, and A. Feringa, “Risk management guide for information technology systems,” NIST Special Publication 800-30, 2002.
- [10] A. Tantawy, S. Abdelwahed, A. Erradi, and K. Shaban, “Model-based risk assessment for cyber physical systems security,” *Comput. Secur.*, vol. 962020, Art. no. 101864.
- [11] A. P. Moore, R. J. Ellison, and R. C. Linger, “Attack modeling for information security and survivability,” Softw. Eng. Inst., Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-2001-TN-001, 2001.
- [12] A. Tantawy, S. Abdelwahed, and Q. Chen, “Continuous stirred tank reactors: Modeling and simulation for CPS security assessment,” in *Proc. 11th Int. Conf. Comput. Intell. Commun. Netw.*, 2019, pp. 117–123.
- [13] K. Stouffer, J. Falco, and K. Scarfone, “Guide to industrial control systems (ICS) security,” NIST Special Publication 800-82, 2011, p. 164.
- [14] A. Tantawy, “Automated malware design for cyber physical systems,” in *Proc. 9th Int. Symp. Digit. Forensics Secur.*, 2021, pp. 1–6.
- [15] S. Mauw and M. Oostdijk, “Foundations of Attack Trees (ser. Lecture Notes in Computer Science), vol. 3935. Berlin, Germany: Springer, 2006, pp. 186–198.
- [16] *NVD—Home*, NIST, Gaithersburg, MD, USA, 2016.
- [17] INTEF and NTNU, *OREDA Offshore and Onshore Reliability Data Volume 1—Topside Equipment*. Bærum, Norway: DNV, 2015.

- [18] *Guidelines for Initiating Events and Independent Protection Layers in Layer of Protection Analysis*. Center Chem. Process Saf., New York, NY, USA, 2015.
- [19] F. Crawley and B. Tyler, *HAZOP: Guide to Best Practice*. Amsterdam, The Netherlands: Elsevier, 2015.
- [20] B. Skelton, *Hazop and Hazan: Identifying and Assessing Process Industry Hazards*. Rugby, U.K.: IChemE, 1999.
- [21] A. M. Dowell, "Layer of protection analysis and inherently safer processes," *Process Saf. Prog.*, vol. 18, no. 4, pp. 214–220, 1999.
- [22] F. Garzia, M. Lombardi, M. Fargnoli, and S. Ramalingam, "PSA-LOPA—A novel method for physical security risk analysis based on layers of protection analysis," in *Proc. Int. Carnahan Conf. Secur. Technol.* 2018, pp. 1–5.
- [23] *Security for Industrial Automation and Control Systems – Part 4-1: Secure Product Development Lifecycle Requirements*, IEC Standard IEC 62443-4-1, 2018.
- [24] H. Kanamaru, "Bridging functional safety and cyber security of SIS/SCS," in *Proc. 56th Annu. Conf. Soc. Instrum. Control Eng. Jpn.*, 2017, pp. 279–284.
- [25] *Nuclear Power Plants—Instrumentation and Control Systems—Requirements for Coordinating Safety and Cybersecurity*, IEC Standard IEC 62859:2016, 2016.
- [26] C. Torens, "Safety versus security in aviation, comparing DO-178C with security standards," in *Proc. AIAA Scitech 2020 Forum*, 2020, Art. no. AIAA 2020-0242.
- [27] A. Avižienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan.–Mar. 2004.
- [28] A. Tantawy, A. Erradi, and S. Abdelwahed, "A modified layer of protection analysis for cyber-physical systems security," in *Proc. 4th Int. Conf. Syst. Rel. Saf.*, Rome, Italy, 2019, pp. 94–101.
- [29] A. J. Kornecki and M. Liu, "Fault tree analysis for safety/security verification in aviation software," *Electronics*, vol. 2, pp. 41–56, 2013.
- [30] C. W. Johnson, "CyberSafety: On the interactions between cybersecurity and the software engineering of safety-critical systems," *Lab. Med.*, vol. 21, no. 7, pp. 411–413, 2012.
- [31] A. J. Kornecki and J. Zalewski, "Safety and security in industrial control," in *Proc. Annu. Workshop Cyber Secur. Inf. Intell. Res.*, 2010, Art. no. 77.
- [32] T. Novak, A. Treytl, and P. Palensky, "Common approach to functional safety and system security in building automation and control systems," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Autom.*, 2007, pp. 1141–1148.
- [33] T. Novak and A. Treytl, "Functional safety and system security in automation systems—A life cycle model," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Autom.*, 2008, pp. 311–318.
- [34] T. Novak and A. Gerstinger, "Safety- and security-critical services in building automation and control systems," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3614–3621, Nov. 2010.
- [35] K. Sørby, "Relationship between security and safety in a security-safety critical system: Safety consequences of security threats," Ph.D. dissertation, Dept. Mech. Ind. Eng., Norwegian Univ. Sci. Technol., Trondheim, Norway, 2003.
- [36] K. Stølen *et al.*, "Model-based risk assessment in a component-based software engineering process," in *Business Component-Based Software Engineering*. Boston, MA, USA: Springer, 2003, pp. 189–207.
- [37] B. Hunter, "Integrating safety and security into the system lifecycle," in *Proc. Improving Syst. Softw. Eng. Conf.*, 2009, p. 147.
- [38] R. Winther, O. A. Johnsen, and B. A. Gran, *Security Assessments of Safety Critical Systems Using HAZOPS* (ser. Lecture Notes in Computer Science), vol. 2187. Berlin, Germany: Springer, 2001, pp. 14–24.
- [39] C. Schmittner, T. Gruber, P. Puschner, and E. Schoitsch, *Security application of Failure Mode and Effect Analysis (FMEA)* (ser. Lecture Notes in Computer Science), vol. 8666. Cham, Switzerland: Springer, 2014, pp. 310–325.
- [40] *GSN Community Standard Version 1*, Origin Consulting, York, U.K., 2011.
- [41] S. Lautieri, D. Cooper, and D. Jackson, "SafSec: Commonalities Between Safety and Security Assurance," in *Constituents of Modern System-Safety Thinking*. London, U.K.: Springer, 2007, pp. 65–75.
- [42] N. Subramanian and J. Zalewski, "Quantitative assessment of safety and security of system architectures for cyberphysical systems using the NFR approach," *IEEE Syst. J.*, vol. 10, no. 2, pp. 397–409, Jun. 2016.
- [43] I. Nai Fovino, M. Masera, and A. De Cian, "Integrating cyber attacks within fault trees," *Rel. Eng. Syst. Saf.*, vol. 94, pp. 1394–1402, 2009.
- [44] M. Steiner and P. Liggesmeyer, "Combination of safety and security analysis—Finding security problems that threaten the safety of a system," in *Proc. 32nd Int. Conf. Comput. Saf., Rel. Secur.*, 2013, pp. 1–8.
- [45] R. Kumar and M. Stoelinga, "Quantitative security and safety analysis with attack-fault trees," in *Proc. IEEE Int. Symp. High Assurance Syst. Eng.*, 2017, pp. 25–32.
- [46] H. Abdo, M. Kaouk, J. M. Flaus, and F. Masse, "A safety/security risk analysis approach of industrial control systems: A cyber bowtie—Combining new version of attack tree with bowtie analysis," *Comput. Secur.*, vol. 72, pp. 175–195, 2018.
- [47] M. Bouissou and J. L. Bon, "A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes," *Rel. Eng. Syst. Saf.*, vol. 82, pp. 149–163, 2003.
- [48] S. Kriaa, M. Bouissou, F. Colin, Y. Halgand, and L. Piètre-Cambacédès, *Safety and Security Interactions Modeling Using the BDMP formalism: Case Study of a Pipeline* (ser. Lecture Notes in Computer Science), vol. 8666. Cham, Switzerland: Springer, 2014, pp. 326–341.
- [49] S. Kriaa, M. Bouissou, and L. Piètre-Cambacédès, "Modeling the Stuxnet attack with BDMP: Towards more formal risk assessments," in *Proc. 7th Int. Conf. Risks Secur. Internet Syst.*, 2012, pp. 1–8.
- [50] B. Kaiser, C. Gramlich, and M. Förster, "State/event fault trees-A safety analysis model for software-controlled systems," *Rel. Eng. Syst. Saf.*, vol. 92, pp. 1521–1537, 2007.
- [51] M. Roth and P. Liggesmeyer, "Modeling and analysis of safety-critical cyber physical systems using state/event fault trees," in *Proc. 32nd Int. Conf. Comput. Saf., Rel. Secur.*, 2013.
- [52] R. Mitchell and I. R. Chen, "Effect of intrusion detection and response on reliability of cyber physical systems," *IEEE Trans. Rel.*, vol. 62, no. 1, pp. 199–210, Mar. 2013.
- [53] C. W. Ten, C. C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for SCADA systems," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1836–1846, Nov. 2008.
- [54] A. J. Kornecki, N. Subramanian, and J. Zalewski, "Studying interrelationships of safety and security for software assurance in cyber-physical systems: Approach based on Bayesian belief networks," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, 2013, pp. 1393–1399.
- [55] G. Sindre, "A look at misuse cases for safety concerns," in *Proc. Working Conf. Method Eng.*, 2007, pp. 252–266.
- [56] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Eng.*, vol. 10, pp. 34–44, 2005.
- [57] C. Raspotnic, P. Karpati, and V. Katta, *A Combined Process for Elicitation and Analysis of Safety and Security Requirements* (ser. Lecture Notes in Business Information Processing), vol. 113. Berlin, Germany: Springer, 2012, pp. 347–361.
- [58] J. Jürjens, *Developing Saf.-Crit. Syst. With UML* (ser. Lecture Notes in Computer Science), vol. 2863. Berlin, Germany: Springer, 2003, pp. 360–372.
- [59] J. Jürjens, *UMLsec: Extending UML for Secure Systems Development* (ser. Lecture Notes in Computer Science), vol. 2460. Berlin, Germany: Springer, 2002, pp. 412–425.
- [60] J. Jürjens, "Developing safety-and security-critical systems with UML," in *Proc. DARP Workshop, Loughborough, U.K.*, 2003.
- [61] G. Pedroza, L. Apvrille, and D. Knorreck, "AVATAR: A SysML environment for the formal verification of safety and security properties," in *Proc. 11th Annu. Int. Conf. New Technol. Distrib. Syst.*, 2011, pp. 1–10.
- [62] J. Thomas, "Extending and automating STPA for requirements generation and analysis," Ph.D. dissertation, Eng. Syst. Division, Massachusetts Inst. Technol., Cambridge, MA, USA, 2013.
- [63] I. Friedberg, K. McLaughlin, P. Smith, D. Laverty, and S. Sezer, "STPA-SafeSec: Safety and security analysis for cyber-physical systems," *J. Inf. Secur. Appl.*, vol. 34, pp. 183–196, 2017.
- [64] A. Nourian and S. Madnick, "A systems theoretic approach to the security threats in cyber physical systems applied to Stuxnet," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 1, pp. 2–13, Jan./Feb. 2018.