

Conflict Analysis and Resolution of Safety and Security Boundary Conditions for Industrial Control Systems

Chidi Agbo

*Computer Science Department
Boise State University
Boise, United States
chidiagbo@u.boisestate.edu*

Hoda Mehrpouyan

*Computer Science Department
Boise State University
Boise, United States
hodamehrpouyan@boisestate.edu*

Abstract—Safety and security are the two most important properties of industrial control systems (ICS), and their integration is necessary to ensure that safety goals do not undermine security goals and vice versa. Sometimes, safety and security co-engineering leads to conflicting requirements or violations capable of impacting the normal behavior of the system. Identification, analysis, and resolution of conflicts arising from safety and security co-engineering is a major challenge, an under-researched area in safety-critical systems(ICS). This paper presents an STPA-SafeSec-CDCL approach that addresses the challenge. Our proposed methodology combines the STPA-SafeSec approach for safety and security analysis and the Conflict-Driven Clause Learning (CDCL) approach for the identification, analysis, and resolution of conflicts where conflicting constraints are encoded in satisfiability (SAT) problems. We apply our framework to the Tennessee Eastman Plant process model, a chemical process model developed specifically for the study of industrial control processes, to demonstrate how to use the proposed method. Our methodology goes beyond the requirement analysis phase and can be applied to the early stages of system design and development to increase system reliability, robustness, and resilience.

Index Terms—Industrial Control Systems(ICS), Cyber-physical systems(CPS), Safety and Security co-engineering, STPA-SafeSec, Conflict-Driven Clause Learning(CDCL)

I. INTRODUCTION

The advancement of technology and the introduction of industry 4.0 have increased complexity in the design and development of cyber-physical systems (CPS). CPS combines both hardware and software resources for computational, communication, and control purposes that are co-designed with the physically engineered components. Examples of CPS include industrial control systems (ICS) that monitor and control physical processes with feedback loops, where physical processes affect computation and vice versa [1]. The deep integration of discrete computing and continuous physical processes poses major engineering challenges, including the co-engineering of safety and security requirements. Safety and security are two important properties of an ICS. During

National Science Foundation Computer and Information Science and Engineering (CISE), award number 1846493 of the Secure and Trustworthy Cyberspace (SaTC) program: Formal TOols foR SaFety aNd Security of Industrial Control Systems (FORENSICS)

the early stages of the design and development of a CPS, safety and security were treated as separate entities. During 2014 and 2015, Chockalingam [2] argued that the research community began to realize the importance of integrating safety and security into CPS. Consequently, research funding and educational projects that focus on safety and security in CPS amount to approximately \$34,000,000 each year in the US [3].

Safety engineering studies the potential accidents¹ of a system that could lead to hazards² and losses. In ICS, losses are classified as acceptable and unacceptable losses. Safety approaches focus on measures to prevent unacceptable losses, such as equipment damage, loss of life, or financial loss. On the contrary, security engineering focuses on mitigating risks³, threats⁴, and vulnerabilities⁵ that exist within the target system. CPS has been classified into safety-critical systems and security-critical systems. In a safety-critical system, safety takes precedence over security and vice versa, with different modeling and analysis tools.

Integrating safety and security into ICS is needed to protect the system safely and securely against potential accidents and attacks. However, the combination of safety and security sometimes introduces a conflict, a challenge that has not been fully addressed by the scientific community or industry. kavallieratos et al. [4] in their comprehensive study of safety and security co-engineering have pointed out that the areas of identifying and resolving conflicting safety and security requirements are under-researched. Conflicting requirements, if not resolved, place an ICS in vulnerable states that can be exploited by cybercriminals.

In 1987, Leveson [5] adopted the use of Petri net modeling over time for the safety analysis of a critical CPS. Other

¹Accidents are unintentional or unplanned events that cause an adverse effect

²Hazard is any condition capable of causing human injury or damage to equipment or to the environment

³Risk is the likelihood that something bad might occur

⁴Threat is a potential risk or vulnerability that can impact the system negatively when exploited by a malicious actor

⁵Vulnerability is any weakness that poses threat to the system

research works that separate safety from security include ([6]–[12]). The concept of integration of safety and security analysis gained deep recognition in 2015 with a focus on unification and integration approaches where safety and security are handled as different units and later merged during the life cycle of system development [2], [13], [14]. In practice, we observed that safety and security co-engineering should be part of the early stages of system design and development.

This paper is directed toward safety and security analysis of ICS, focusing on safety and security co-engineering, identification, and resolution of conflicting constraints that expose the system to failure or attacks. The overall goal is to increase the reliability, robustness, and resiliency of an ICS. To this end, we propose the integration of the STPA-SafeSec approach for safety and security analysis and the Conflict-Driven Clause Learning (CDCL) approach for the identification, analysis, and resolution of conflicts. The reasons for our approach are (1). STPA-SafeSec is a systematic technique that adopts a top-down approach to perform an integrated safety and security analysis that captures abnormal behaviors of the system due to component interaction failure. (2). The CDCL approach has been shown to be the best modern SAT technique for the detection, analysis, and resolution of conflicts.

Our work focuses mainly on the analysis and resolution of conflicts that arise from bounded constraints in safety and security and does not cover all aspects of conflicts that exist in the integration of safety and security, such as confidentiality and/or availability-related conflicts, with a Python script for our implementation. The rest of this paper is organized as follows. Section II discusses related work. Our proposed methodology is presented in Section III. We bring our methodology to the Tennessee Eastman Plant Process in Section IV. Conclusions and future work are discussed in Section V.

II. RELATED WORK

Kornecki et al. [15] provided a mutual relationship between safety and security in industrial control systems. Their work focused on protecting the software system and the operating environment by proposing the adoption of a safety shell in building security programs. Bruner et al. [16] proposed an integrated model for CPS safety and security requirements based on requirement engineering, risk management, and evidence documentation to support the re-certification process. Bamberger and Martin [17] proposed a process for integrating safety and security according to ISO 26262 standards. ISO 26262 framework provides guidelines for developing safe automotive applications. Cimatti et al. [18] work goes beyond safety and security integration to develop an approach to verifying safety and security requirements to detect system dependencies and failures.

The analysis and management of the safety risks were performed independently from the security counterpart. However, as the interconnectivity and complexity of CPS increased, the research community recognized the need to combine the analysis of safety and security-related risks. Hayakawa et al. [19] proposed a risk analysis method that deals with safety and

security based on the STPA approach. They applied this technique to a medical device (insulin pump) for diabetic patients to uncover accidents that cannot be prevented by functional safety. Kriaa et al. [20] present an S-cube approach to joint risk assessment that is useful for different stages of system development. Reichenbach et al. [21] proposed an approach that extended threat vulnerability and risk assessment (TVRA) for integrated security and safety risk analysis. Gleirscher et al. [22] identified technical and sociotechnical challenges in the coassurance of safety and security for collaborative industrial robots (Cobots) without specifying approaches to identify, analyze and resolve risks / conflicts related to safety and security. For example, no testing approach was proposed for Cobots to minimize their overall safety and security risks.

Furthermore, the majority of research work done in this area focused mainly on safety and security co-engineering in ICS with very little work on how to identify and resolve conflicts that arise as a result of their integration. Menon and Vidalis [23] present various CPS domains where safety and security conflicts can exist in the system domains, but that was just a pointer. They suggested an in-depth security approach for conflict resolution that relies on providing multiple layers or overlapping safety and security mitigations, but this method can introduce redundancy and/or more conflicts in the system. Gu et al. [24] proposed an approach for the analysis of safety and security requirements and conflict resolution in industrial control systems. They used risk-based scoring provided by domain experts for conflict resolution, which can be misleading, as the impact of an action or event can sometimes be underestimated, leading to low scoring. Furthermore, this approach cannot be applied to unrelated safety and security requirements or design and implementation phases. Farooq et al. [25] present a formal approach to identify conflicting actions between the controller and actuators in an IoT environment. In Sun et al. [26], they proposed a methodology to find conflicting requirements in automatic door systems. Other related research focused on the automotive and aviation industries with little focus on ICS. They include ([27], [28], [29], [30]). Our work is different in that it goes beyond the requirement analysis phase to system design and implementation using a case study that reflects a real-world ICS. In addition, we provide a systematic conflict analysis and resolution approach that uses the concept of a conflict-driven clause learning approach.

III. METHODOLOGY

In this work, we propose a methodology based on the STPA-SafeSec-CDCL approach that integrates the STPA-SafeSec [31] and conflict-driven clause learning (CDCL) [32] approaches for the analysis of safety and security, the detection of conflicts, and resolution. This approach applies to the analysis of safety and security in the requirement phase, the design of the system, or the implementation phase.

Figure 1 represents our proposed methodology applied to our case study in the next section.

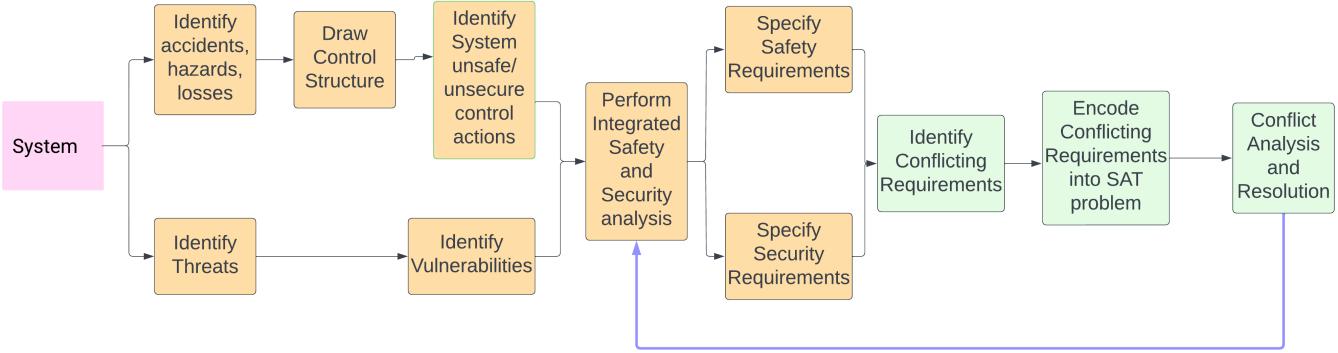


Fig. 1: Proposed Methodology

A. The STPA-SafeSec-CDCL Approach

The STPA-SafeSec technique was developed on top of the STPA [33] and STPA-Sec approaches to address the deficiencies of STPA and STPA-Sec that perform safety and security analysis separately. STPA-SafeSec provides an approach to integrated safety and security analysis from a top-down perspective. Detects both component failures and component interaction failures. Identifies security vulnerabilities⁶ and requirements⁷ such as scenarios that lead to violation of safety and security constraints, and uses the results to refine system concepts⁸ to be safer and secure [31]. Furthermore, STPA-SafeSec maps causal scenarios or vulnerabilities to losses and ensures the development of appropriate controls (constraints) over the behavior of the system [34]. On the other hand, conflict-driven clause learning (CDCL) is an algorithm for solving Boolean satisfiable (SAT) problems with great capabilities in conflict analysis and resolution. Our proposed methodology combines the concept of STPA-SafeSec and the CDCL approach for co-engineering of safety and security, conflict detection, and resolution. One of the major advantages of our proposed methodology over other methodologies is the fact that it takes into consideration the conflicts or identification of conflicts that may arise from safety and security co-engineering and/or from system design and implementation phases.

The STPA-SafeSec-CDCL methodology consists of six phases. The first three phases focus on a detailed analysis of safety, security, and their integration. Other phases consist of identifying, analyzing, and resolving conflicts that exist during integration, thus specifying mitigation strategies. In this section, we will discuss in detail each of the six phases of our proposed framework.

1) *Perform In-depth Safety Analysis*: This is the first phase of the STPA-SafeSec-CDCL approach. In this phase, the con-

⁶Security vulnerabilities are deficiencies, errors or faults found within the system security

⁷Security requirements are protection capabilities that need to be satisfied in order to achieve the security of a system

⁸System concept denotes the integration and interdependencies of system components and their interactions.

trol structure of the system is analyzed in order to identify and specify possible control actions and/or unsafe control actions. This phase enables domain experts to identify and classify accidents or hazards that lead to acceptable or unacceptable losses as presented in Table I. Phase 1 allows for the mapping of control actions to the component layer or the process variables they control. This phase is described in detail in [35].

2) *Perform In-depth Security Analysis*: This phase focuses on the identification and classification of various threats and vulnerabilities to the system. In-depth security analysis tends to analyze and uncover threats related to the confidentiality, integrity, and availability of the system. In addition, this phase equips security experts with the necessary knowledge about the system to find weaknesses that can threaten the security of the system. Examples of such threats include command injection, command manipulation, command delay or drop, measurement drop or manipulation, etc. as explained in section IV.

3) *Perform Integrated Safety and Security Analysis*: Here, safety and security experts come together to ensure that security goals enhance safety and do not undermine safety goals and vice versa. Phase 3 is crucial to ensure a safe and secure ICS. Enhances the harmonization of safety and security goals toward the overall safety and security of the system. safety and security requirements are specified based on accidents, hazards, losses, causal factors, unsafe control actions, threats, and vulnerabilities identified in Phases 1 and 2.

4) *Identification of Conflicting requirements*: The integration of safety and security analysis can introduce conflicts that are known or unknown to safety and security experts. This phase can be the most difficult, depending on the system being analyzed. Sometimes there are conflicts within the safety and security domain or their integration. However, design or implementation goals can conflict with safety or security goals. For example, consider system A designed to operate properly under noise with no significant impact on the system. However, the introduction of noise at a certain

state of the system can cause an increase or decrease in some process variables in A that conflict with the safety or security constraints or the normal behavior of system A. During the conflict detection phase, all possible scenarios or causes that could lead to a conflict are analyzed to ensure that the system operates in a conflict-free manner. In this paper, we adopt a conflict identification approach proposed by Thomas [36] that is based on four principles (i) source controller issuing control actions (SC), (ii) type of control action (T) such as provided or not provided, (iii) control action (CA) and (iv) context in which control action is provided or not provided (Co). A conflict occurs when providing or not providing a control action leads to hazards/threats. For example, consider an automatic door system(ADS) used in banks, airports, and other critical facilities designed to detect metal objects(e.g., guns) and automatically deny the person an entrance into the building. The security requirement for such a system is designed to ensure that the ADS shuts the door against anyone with metal objects. However, in the case of emergencies, such as fire outbreaks, the ADS will enforce safety requirements for evacuation purposes, thereby compromising security goals. In this case, enforcing or not enforcing safety/security requirements leads to hazards/threats. We add this conflict identification method to our case study in section IV.

5) *Conflict Analysis and Resolution:* Conflict analysis and resolution are conducted when there are conflicting requirements that occur during integration of safety and security or when design goals conflict with safety/security goals. This phase enables domain experts to identify causal factors and conflict resolution strategies. The knowledge gained is needed to redefine the safety and security constraints or the goals of system design. Our methodology adopts the CDCL approach to conflict analysis and resolution discussed in detail in section III-C

6) *Redefine Safety and Security Constraints, and/or Mitigation strategies:* In this phase, safety and security requirements specified in phase 3 are re-evaluated to meet the system's current state based on the knowledge gained from the conflict analysis and resolution phase. Safety and security constraints are redefined to ensure conflict-free constraints. Mitigation strategies can also be specified to address the overall safety and security goals of the system.

B. STPA-SafeSec Approach

The STPA-SafeSec approach is based on the fundamental principles of STPA with an extension to the security domain. The STPA-SafeSec approach addresses the weaknesses of the STPA and STPA-sec frameworks by providing interdependencies between security and safety constraints. The result of the STPA-SafeSec approach helps domain experts identify potential hazards (safety accidents) or threats(security vulnerability) that can cause system losses [37]. Although STPA-sec shows that STPA can also be used for security analysis to ensure system safety [31], [38]. However, the STPA approach prioritizes safety over security. On the contrary, the STPA-

SafeSec provides a unified and integrated approach that gives equal relevance to both the safety and security of the system and the reason for its use in this paper. In STPA-SafeSec, safety and security constraints are specified based on the control structure of the system. Furthermore, in the STPA-SafeSec analysis, we first identify system losses and hazards, control structures, threats, and / or vulnerabilities, and safety and security constraints. We applied this approach to our case study in section IV.

C. Conflict Driven Clause Learning Approach (CDCL)

The CDCL method is similar to the DPLL approach with the standard backtracking search procedure where unit propagation is executed after each decision assignment, creating a new decision level [32]. The CDCL approach follows a non-chronological backtracking search technique in which backtracking occurs once a conflict is identified, and during conflict resolution, new clauses are learned. The CDCL approach jump starts at the appropriate decision level during backtracking when conflicts occur. The CDCL approach operates on clauses in the form of a conjunctive normal form(CNF). Clauses are literals. A literal is a variable a or its negation $\neg a$. A CNF represents a conjunction of disjunctive clauses⁹. For example, consider a set of literals $a, b, \neg a$ that takes a true (1) or false(0) value. Clauses $(a \vee b)$ and $(\neg a \vee b)$ are of disjunctive form. Therefore, a CNF denotes clauses of the form $(a \vee b) \wedge (\neg a \vee b)$. **Definition-1 Conflict:** Consider a CNF formula $\psi = (a \vee b) \wedge (\neg a \vee b)$ such that an assignment of true value to variable a and false to variable b leads to an unsatisfiable result (UNSAT). A conflict occurs when a variable a or b is assigned both true and false values in a given formula. A CDCL solver returns the UNSAT result if and only if it is unable to resolve the conflict by finding an assignment that satisfies the formula and otherwise returns a SAT result. During the assignment of variables, CDCL adopts the concept of unit propagation or Boolean constraint propagation(BCP) to determine the Boolean value that must be assigned to a particular variable for a formula to be satisfied. For example, consider the formula $\alpha = (a \vee b \vee c \vee d)$. If we assign a false value to variables a, b , and c , by unit propagation or BCP, CDCL will assign the true value to variable d for α to be satisfiable. If it is a unit clause, CDCL will assign a true value to the variable for it to be satisfiable.

Furthermore, in the case of a conflict, CDCL backtracks to the appropriate decision level made before the conflict occurred. During conflict analysis and resolution, implication graphs are created to help analyze and visualize the assignment of variables or decisions that caused a conflict. It is a powerful graph for conflict analysis and resolution, as we can easily backtrack to the last decision made before a conflict, undo the assignment, and reassign variables again until the conflict is resolved. During conflict resolution, new clauses are learned. The modern CDCL SAT solver implements the concept of the first unique implication point (UIP) for clause learning. UIP

⁹Disjunctive clauses are clauses consisting of OR-ed literals

represents the closest node or path to the conflicting node in the implication graph. The use of first UIPs is necessary to reduce the learning of new redundant clauses. Let C_1 and C_2 represent the clauses $(a \vee b)$ and $(\neg a \vee b)$ in ψ , respectively. Then, the implication graph for the CNF formula ψ is given in Figure 2a.

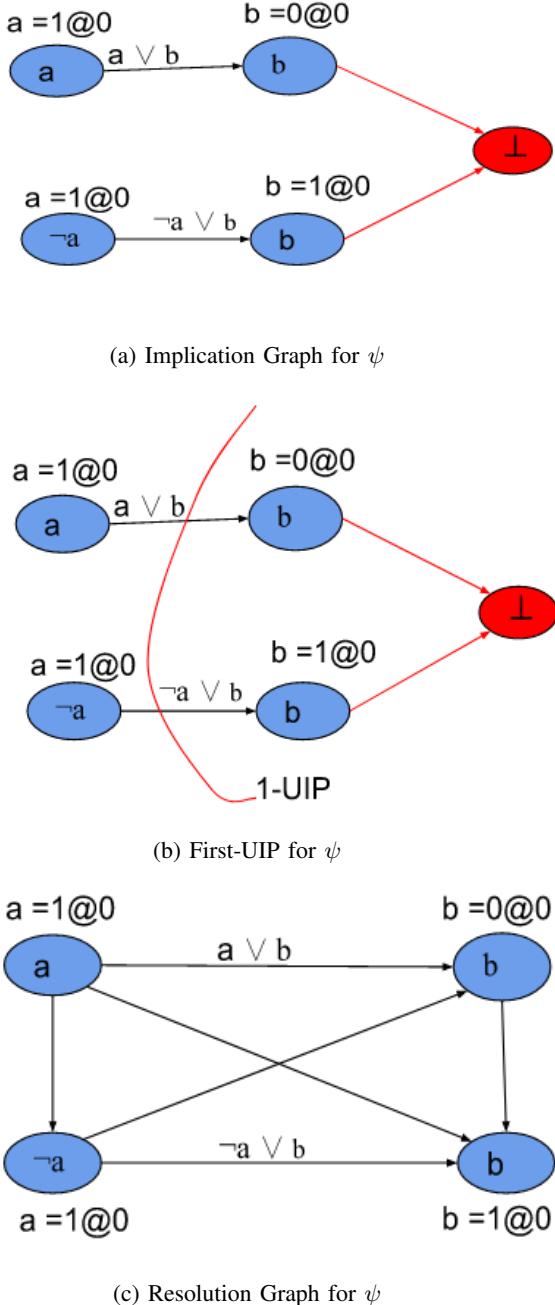


Fig. 2: Conflict Analysis and Resolution for formula ψ

In the implication graph, as shown in Figure 2a, we assign true(1) and false(0) values to variables \mathbf{a} and \mathbf{b} , respectively, at a decision level 0 for C_1 to be satisfiable. If we maintain the same assignment for C_2 , C_2 becomes unsatisfiable, thus

causing the general formula ψ to return the UNSAT result. Therefore, to satisfy ψ , we need to assign a true (1) value to the variable \mathbf{b} that contradicts the initial assignment of $\mathbf{b} = \text{false}(0)$ leading to a conflict. Once a conflict occurs, CDCL jumps back to the decision level that led to the conflict. In our case, we go back to decision level 0 and undo the assignment of \mathbf{a} . During the backtracking, new clauses are learned. The 1-UIP as shown in Figure 2b helps us to make the right cut for the nodes closest to the conflicting nodes. In our example, the learned clause C is $(a \vee \neg a)$. The new clause is called a conflict clause. The conflict clause is added to formula ψ to avoid repeating the same assignment that leads to a conflict. During conflict resolution (Figure 2c), we reassigned the false value (0) to \mathbf{a} and the true value (1) to \mathbf{b} to make ψ satisfiable.

Safety and security requirements in ICS are often represented as bounded constraints, where the system and its process variables are expected to be within a set of intervals and not as a Boolean or CNF formula. Therefore, one of the intriguing parts of this section is how to encode conflicting requirements into SAT problems. Therefore, we propose a method (see Figure 3) to encode bounded constraints in a SAT problem.

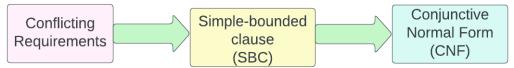


Fig. 3: Encoding Conflicting requirements into SAT problem

In this method, the safety or security requirements are first converted to a simple bounded clause (SB) and then to a CNF. The concept of SB has been applied in the areas of linear measurement [39], stochastic systems for calculating the fluctuation of time-based currents [40], nonlinear control systems for computing a set of reachable states based on differential inequalities [41], interval-based arithmetic constraints [42], machine learning for computing simple bounds for interval-valued covariates in linear regression [43], artificial intelligence for defining and controlling sub-spaces of a network of agents [44], etc. Our encoding method is similar to the work done by [45]. To the best of our knowledge, our work is the first to lift the concept of a simple bound to safety and security constraints in ICS for the identification of conflicts and resolution. We have implemented encoding of conflicting requirements for bounded constraints in Python¹⁰ and it is used in our case study.

IV. CASE STUDY

The Tennessee Eastman challenge process (TEP) is the simulation of a real chemical process developed specifically for the study of industrial control processes. We choose the TEP plant for 3 main reasons (i) The TEP plant is a widely used plant process model for the study of CPS [46], [47]. (ii) It is made up of various components, levels, and process variables found in today's chemical plants such as reactor,

¹⁰<https://github.com/Chidi93/IEEE6thConfItaly.git>

compressor, stripper, condenser, separator, analyzers, sensors, actuators(valves), feed components, pressure, temperature, etc. and (iii) TEP has been used in the study of CPS security and attack detection [48], [49], [50], [51], [52], etc.

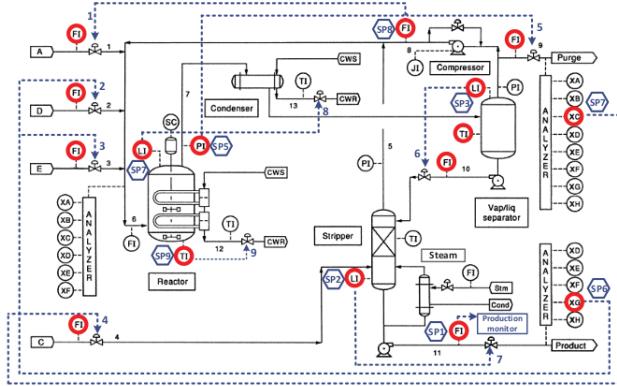
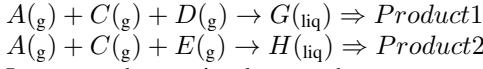


Fig. 4: The Tennessee Eastman Plant [53]

The plant contains eight chemical components comprising four reactants (A, C, D, and E), two liquid products, one by-product, and one inert component as seen below;



In our study, we implemented our proposed methodology -the STPA-SafeSec-CDCL framework - in the TE Chemical reactor control system (CRCS) described in [9]. The TE chemical reactor is the unit of the TE plant where the chemical reactions for the creation of products G and H take place. In the STPA-SafeSec-CDCL approach, the first phase is to perform an in-depth safety analysis of the CRCS system to identify losses and hazards at the system level (see Table I).

The next step is to examine the control structure to learn the detailed list of commands, manipulated process variables (MPV), and feedback during chemical production.

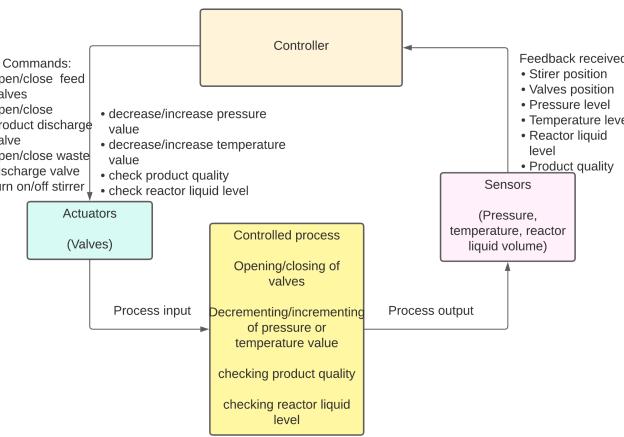


Fig. 5: TE CRCS

The goals and objectives of the TE control system have been described by down and Vogel [54] as follows: (i) keeping

TABLE I: Losses and Hazards associated with the TE Chemical reactor

Losses	Reference	Refers-To
Loss of life or injury	[L-1]	
Loss of or damage to equipment	[L-2]	
Loss of or damage to product	[L-3]	
Monetary loss	[L-4]	
Contaminated environment	[L-5]	
Hazards		
Operating reactor beyond set points	[H-1]	[L-1],[L-2],[L-3],[L-4],[L-5]
Plant inability to maintain process variables within defined threshold during chemical production	[H-2]	[L-1],[L-2],[L-3],[L-4],[L-5]
Release of high volume of chemicals into the reactor	[H-3]	[L-1],[L-2],[L-3],[L-4],[L-5]
Opening reactor's product and discharge valves at the same time	[H-4]	[L-3],[L-4]
Plant releases contaminated materials	[H-5]	[L-1],[L-5]

the process variables within the desired values. (ii) ensure that the processes operate within equipment constraints. (iii) Reduce the variations in product rate and process variables during disturbances, (iv) Reduce valve movement that affects other processes, and (iv) recover quickly and smoothly from disturbances, product mix, or production rate changes.

We analyze the CRCS system to identify necessary control actions (CA) and unsafe control actions (UCA) that cause hazards that lead to loss. We applied the four ways of identifying CA and UCA (see Table II) described by Leveson and Thomas [35].

A. Safety Requirements

As shown in Table II, we can derive unsafe control actions and their causes(CS) for each control action.

CA-1:Equipment must be operated within set points

UCA-1:Equipment is operated beyond set limits.

CS-1:Controller fails to regulate the operation of the plant within set points.

CA-2:Maintain process variables within desired values

UCA-2:Operating plant under out-of-bounds process variables

CS-2.1:Failure of sensors to properly monitor the current state of the system

CS-2.2:Communication gap between sensor and controller.

CA-3:Minimize frequent movement of the valves

UCA-3:Opening/closing of valves without restraint

CS-3.1:Controller issue a command that causes the opening

TABLE II: CRCS Control Actions and their Impact on the System when providing, not providing, providing too early / late or stopping too early or late

Control Action	Not providing causes hazard	Providing causes hazard	Provided too early/late	Stopped too early/late
Equipment must be operated within set points (CA-1) leads to shutdown or equipment or product damage Minimize variability in valves (CA-3) Reduce process variables and feed rates variation under disturbances (CA-4) The plant must recover quickly and smoothly from disturbances and production rate changes (CA-5)	Inability to control reactor operations within set limits leads to shutdown or equipment damage N/A High release of raw materials into the reactor causing reactor overflow or product damage causes system shutdown or product damage system shutdown, equipment or product damage	N/A Too late causes equipment shutdown/damage or product damage Opening of product and waste valves at the same time causes reactor overflow or product damage N/A N/A	Enforcing operating equipment within set points but too late causes damage to equipment or worn out N/A Closing of the feed or discharge valves too early/late impact the product quality too late leads shutdown or product damage too late causes system shutdown, equipment or product damage	Stopping too late when equipment operating out of bound causes damage Maintain process variable within desired values (CA-2) Opening/closing of the feed valves for too long/early impacts product quality Stopped too early/late causes shutdown or product damage stopped too early / late leads to system shutdown, equipment or product damage

of both product and waste valves at the same time due to poor feedback from the sensor.

CS-3.2:Sensor's failure to communicate the current state of the valves to the controller.

CS-3.3:Actuator's failure to receive commands from the controller.

CA-4: Reduce the variation in feed rates under disturbances

UCA-4:Activation of disturbances that increase feed rates

CS-4: Control system inability to stabilize feed rates under disturbances due to failure or attack. **CA-5:** Plant must recover quickly and smoothly from disturbances **UCA-5:** Plant does not recover under disturbances. **CS-5:** Increase in the variable of the process under disturbances that cause the plant to shut down.

B. Security Requirements

Having performed a detailed system safety analysis, we can now perform the security analysis. The security analysis phase is based on security threats(SCT) and vulnerabilities of the target system. In the TE CRCS, we have identified various threats based on the CIA triad-Integrity, Confidentiality, and Availability.

i) Confidentiality threats

- **SCT-C-1:** Unauthorized access to plant's proprietary data
Scenarios/possible attacks: An external person or a disgruntled employee has access to the operational data of the plant to exploit the plant.
- **SCT-C-2:**Unauthorized access to the HMI or control commands.
Scenarios/possible attacks: Remote connection to the

HMI or control commands by a malicious actor to manipulate the system.

ii) Integrity threats

- **SCT-I-1:**Command Manipulation
Scenarios/possible attacks:Remote connection to the controller or HMI to inject or modify control commands.
- **SCT-I-2:**Command drop
Scenarios/possible attacks:Man in the middle attack (MITM) that drops control commands.
- **SCT-I-3:**Measurement Manipulation
Scenarios/possible attacks:MITM attack that intercepts and modifies sensor measurement with fake values.
- **SCT-I-4:**Measurement drop
Scenarios/possible attacks:Man in the middle (MITM) attack that drops sensor measurement.

iii) Availability threats

- **SCT-A-1:**Command delay
Scenarios/possible attacks: A DOS attack that determines the optimal time to delay control commands and thus drives the plant into an unsafe state.
- **SCT-A-1:**Measurement delay
Scenarios/possible attacks: A DOS attack that delays sensor measurement so that the controller uses stale data from its memory that do not reflect the current state of the plant to make decisions.

C. Safety and Security co-engineering

Safety and security are two essential properties in ICS, and the importance of their integration cannot be overemphasized. Integration of safety and security analysis is necessary to

ensure that safety goals do not undermine security goals and vice versa. When we integrate our results into phases 1 and 2, we can see that unsafe control actions and causes in the Safety domain are synonymous with the identified threats in the Security domain. This phase bridges the gap that exists when the safety analysis is done separately from the security analysis. Safety and security experts collaborate to ensure the overall safe and secure operation of the system. Security experts match threats with possible UCA and their causes to ensure that all potential threats to the system are covered. Security experts re-evaluate the system with safety lenses and vice versa. In this work, we focus on those safety requirements of the TE plant that are represented as bounded constraints. Table III summarizes the safety limits.

TABLE III: Safety Constraints of the TE Plant

Safety Boundaries/Constraints			
Component	Boundary(Feed rates in %)	Component	Boundary(Feed rates in %)
Feed A	[24, 30]	Reactor pressure	[2800, 3000]
Feed C	[60, 62]	Stripper level	[46, 54]
Feed D	[62, 64]	Reactor Level	[54, 55]
Feed E	[52, 55]	Quality	[54, 55]
Product G	[52, 56]	Price	[100, 120]
Product H	[42, 46]	Production	[22, 23]

Defining safety requirements in ICS as bounded constraints uses a strict boundary approach. The reason for this approach is crucial in the safety and security domain for the rapid detection of out-of-bounds constraints. For example, if A is a component, the bound $A \in [25, 30]$ is tighter than $A \in [0, 30]$ and provides more information on the interval of A. Furthermore, there is a violation or conflict when there are actions or events that cause the system to operate outside the set boundaries. In the case of the TE plant, the control system is designed to ensure that the system operates within the defined set points. For example, the TE plant shuts down when the reactor pressure exceeds 3000 kpa.

While safety helps prevent failure, security protects the system against attacks. Some research work has shown the impact of DOS or integrity attacks on the TE plant and how they successfully attacked the system ([53], [55], [56]). On the basis of our analysis, we define the following security requirements to address all identified threats to the system.

i) Confidentiality

SC-C-1: Only authorized users should access the HMI using strong passwords and multifactor authentication for login. Security tools and monitoring systems must be configured to monitor and prevent attempted logins.

SC-C-2: Plant proprietary data must be encrypted to prevent users from reading the plant's operational commands and workabilities.

ii) Integrity

SC-I-1: Every input from the HMI or controller must be validated

SC-I-2: The system should establish an average time required for the controller to communicate with sensors and actuators and vice versa. **SC-I-3:** Build a defensive mechanism to prevent attackers from modifying control commands, sensor, actuator values or feed rates. **SC-I-4:** Every user must be authenticated before accessing plant facilities or systems.

iii) Availability

SC-A-1: Monitor the traffic of the TE network to identify unusual traffic patterns or unusual communication delays between the controller and sensors / actuators

SC-A-2: Make the TE network resistant to DOS attacks by establishing firewalls and placing servers in different data centers.

Integration of safety and security analysis can result in conflicting requirements. Based on our adopted conflict identification approach, in our case study, we found that the activation of some disturbances conflicts with the safety/security and typical behavior of the system, as detailed in the next section. From our analysis, we observed that providing or not providing control action 4(CA-4) leads to hazards/threats under disturbance activation because the controller could not maintain the process variables and feed rates variations within the acceptable threshold, thus conflicting with the plant's safety and security goals.

D. Conflicts Analysis and Resolution:

The introduction of disturbances in chemical plants, such as the TE process model, exposes the plant to noise and variations that could negatively impact the system. The TE plant is designed to operate under disturbances without noticeable effects on the processes, process variables and / or the plant based on the control objectives. Sometimes designers build systems that can withstand noise or uncertainty as a way to prove the reliability, robustness, and resiliency of the system. However, such a design and implementation could lead to redundancy and violations(conflicts) of system properties. In the TE plant, we closely observed the operation of the plant under disturbances and found some redundancy and violations in the implementation.

From our analysis, we found that activation of some disturbances violates the safety and normal behavior of the TE plant. To demonstrate the impact of disturbances on the system, we used the TE plant code developed by Bathelt and Ricker [57]. We run the code in Matlab. In the simulation setup, we set the time T for the simulation at 50 h. First, we run the simulation without disturbance activation using the base case values specified by Downs and Vogel [54] and observe the normal behavior of the plant (see Table III). Using the same setup, we performed other simulations with activation of disturbances IDV(1), IDV(11), and IDV(13) and recorded plant behavior. IDV(1) is a step-type disturbance that manipulates the feed ratios A and C. IDV(11) is a random variation disturbance that

affects the temperature of the reactor cooling water inlet, and IDV(13) is a slow drift-type disturbance that affects reaction kinetics. The specified control actions ensure that the Plant operates properly under disturbances.

Unfortunately, from the simulation result, we observed that disturbance activation conflicts with the safety and security goals of the plant, causing the plant to exhibit abnormal behavior. For example, the production cost increased more than 150% higher than the normal production cost. There was a significant decrease in the production rate and a great increase in the feed rates. The reactor pressure and stripper recorded high and low peak levels, respectively, leading to the shutdown of the plant. Table IV summarizes the impact of disturbance activation on the system. It is important to note that such fluctuations can have catastrophic consequences in chemical, nuclear or water treatment plants.

TABLE IV: Plant production under disturbance activation

Chemical production under disturbances IDV(1)			
Component	Boundary(Feed rates in %)	Component	Boundary(Feed rates in %)
Feed A	[28, 100]	Reactor pressure	[2760, 2820]
Feed C	[55, 61]	Stripper level	[30, 70]
Feed D	[62, 64]	Reactor Level	[62, 68]
Feed E	[52, 55]	Quality	[54, 55]
Product G	[52, 56]	Price	[50, 250]
Product H	[42, 46]	Production	[22, 23]
Chemical production under disturbances IDV(11)			
Component	Boundary(Feed rates in %)	Component	Boundary(Feed rates in %)
Feed A	[28, 100]	Reactor pressure	[2780, 2960]
Feed C	[57, 61]	Stripper level	[-30, 50]
Feed D	[63, 64]	Reactor Level	[64, 69]
Feed E	[53, 60]	Quality	[54, 58]
Product G	[54, 58]	Price	[50, 300]
Product H	[37, 44]	Production	[20, 23]
Chemical production under disturbances IDV(13)			
Component	Boundary(Feed rates in %)	Component	Boundary(Feed rates in %)
Feed A	[10, 45]	Reactor pressure	[2500, 2900]
Feed C	[60, 63]	Stripper level	[10, 80]
Feed D	[63, 64]	Reactor Level	[64, 69]
Feed E	[52, 56]	Quality	[50, 57]
Product G	[54, 58]	Price	[40, 300]
Product H	[37, 44]	Production	[20, 22]

Having identified conflicting security and safety boundaries in our system, as presented in Table IV, we applied the encoding method described in our methodology. We first encoded the constraints into a simple bound clause and then

translated the SB-clause into CNF solvable by the CDCL approach. For example, let D_0 be a domain that represents the overall chemical reactions in the TE reactor that produce the products G and H with or without disturbances. Let D_1 be a subdomain that represents the chemical reactions in the reactor that produce the product G under normal plant operation (without disturbances) such that

$$D_1 = A_{(g)} + C_{(g)} + D_{(g)} = G_{(liq)} \quad (i)$$

Let D_2 be a subdomain that represents the chemical reactions in the reactor that produce the product H under normal plant operation (without disturbances) such that:

$$D_2 = A_{(g)} + C_{(g)} + E_{(g)} = H_{(liq)} \quad (ii)$$

Let D_3 and D_4 be two subdomains that represent the chemical reactions in the reactor that produce products G and H, respectively, under disturbances such that: $D_3 = \neg A_{(g)} + \neg C_{(g)} + \neg D_{(g)} = \neg G_{(g)}$ (iii)

$$D_4 = \neg A_{(g)} + \neg C_{(g)} + \neg E_{(g)} = \neg H_{(g)} \quad (iv)$$

Note that the negation of a component denotes that the lower limit or upper limit of the component or both limits do not fall within acceptable safety limits or threshold.

Now, let us decompose the components that make up D_1 , D_2 , D_3 , and D_4 with their respective limits. For D_1 , we have that:

$$A \in [24, 30], C \in [60, 62], D \in [62, 64]$$

Let l_1, l_3, l_5 , and l_7 represent the lower bounded safety properties, and l_2, l_4, l_6 , and l_8 represent the upper bounded safety properties such that:

$$D_1 = ((a \geq 24) \wedge (a \leq 30)) \vee ((c \geq 60)$$

$$\wedge (c \leq 62)) \vee ((d \geq 62) \wedge (d \leq 64)) \text{ and}$$

$$l_1 \leftrightarrow ((a \geq 24), l_2 \leftrightarrow ((a \leq 30), l_3 \leftrightarrow ((c \geq 60), l_4 \leftrightarrow ((c \leq 62), l_5 \leftrightarrow ((d \geq 62), l_6 \leftrightarrow ((d \leq 64)$$

So, we have:

$$D_1 = (l_1 \wedge l_2) \vee (l_3 \wedge l_4) \vee (l_5 \wedge l_6) \quad (1)$$

$$D_2 = (l_1 \wedge l_2) \vee (l_3 \wedge l_4) \vee (l_7 \wedge l_8) \quad (2)$$

$$D_3 = (l_1 \wedge \neg l_2) \vee (\neg l_3 \wedge l_4) \vee (l_5 \wedge l_6) \quad (3)$$

$$D_4 = (l_1 \wedge \neg l_2) \vee (\neg l_3 \wedge l_4) \vee (l_7 \wedge \neg l_8) \quad (4)$$

Therefore, $D_0 = (D_1 \vee D_2) \wedge (D_3 \vee D_4) \equiv (D_1 \wedge D_3) \vee (D_2 \wedge D_4)$

We pass D_0 to our Python script¹¹ that converts it to a CNF formula. The script converts the individual domains to CNF. It also contains methods for creating implication graphs. The script takes a formula with variables A, B, C, ..., H which represents $l_1, l_2, l_3, \dots, l_8$ (in our case) and converts it to a CNF formula.

During conflict analysis, we use an implication graph. In our analysis, we created implication graphs for IDV (1), IDV (11) and IDV (13). It is imperative to note that the assignment of values to variables l_1, l_2, \dots, l_8 is such that a true value denotes a lower bound or an upper bound within the safety boundary, otherwise false. We adopted this assignment method on the basis of knowledge of the underlying system.

We denote conflicting bounds in red, as seen in Figure 6. A conflicting safety bound occurs when a variable representing

¹¹see details here: <https://github.com/Chidi93/IEEE6thConfItaly.git>

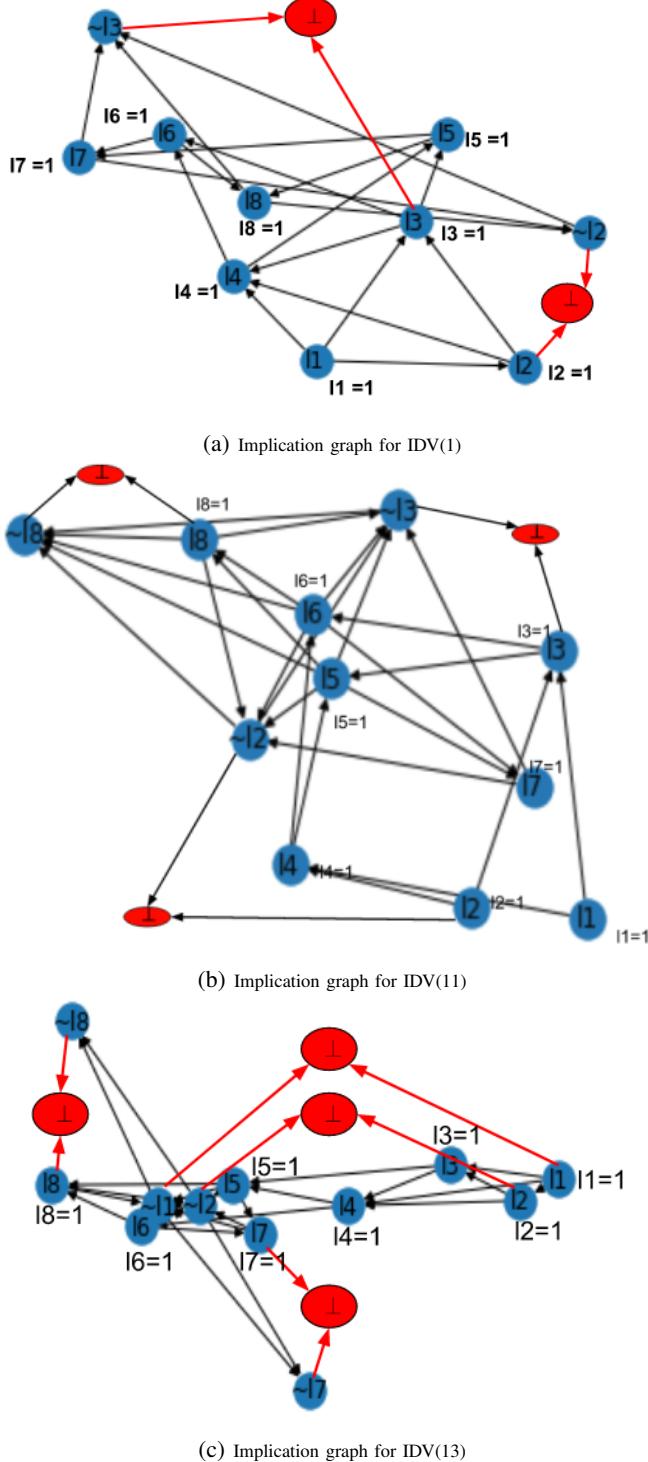


Fig. 6: Conflict Analysis and Resolution Process

a lower or upper bound becomes true and false in the same domain as seen from the implication graph. The implication graph shows which variable is out-of-bound under the activation of disturbances. For example, the variable I_2 represents the upper limit of the amount of feed A that should be in the

reactor per hour. There is a violation of I_2 from 30% to 100% with the activation of IDV(1), IDV(11) and IDV(13) resulting in a conflict. During the conflict analysis and resolution phase, the implication graph helps safety/security and engineering experts identify the main causes of violation of safety/security constraints and/or design goals to make an informed decision.

From our conflict analysis and resolution, we found that the TE plant, although designed to operate properly under disturbances, goes into an unsafe state during the activation of certain disturbances. For example, the activation of IDV (1), IDV(11) and IDV(13) violates the bounded safety constraints of the plant during chemical production. The result of our analysis shows that disturbances 1, 11, and 13 should not be activated during chemical production of TE. This information and knowledge about the plant are necessary to redefine safety, security, or design goals to make the system more reliable, robust, and resilient to attacks.

V. CONCLUSION AND FUTURE WORK

This paper proposes a methodology to address the concept of co-engineering of safety and security in ICS. In this work, we analyze the safety and security requirements of the Tennessee Eastman (TE) plant, where the safety requirements were represented as bounded constraints. The main goal of the paper is to identify, analyze, and resolve any conflicts/violations or vulnerabilities that occur during the safety and security analysis that an adversary can exploit. An attacker with proper knowledge of the system can conduct a well-orchestrated stealthy attack against the system by activating the right disturbances that can impact the system's behavior. Our study is relevant to ICS and other cyber-physical systems (CPS) fields with the aim of increasing the reliability, robustness, and resiliency of these systems against cyber attacks. In our result, we identified and analyzed the conflicts that arise from the safety and security boundary conditions in the TE plant during chemical production. Our analysis covers system integrity-related conflicts and/or violations, but not confidentiality-related and / or availability-related conflicts.

In the future, we will extend our proposed methodology to unbounded safety and security constraints, focusing mainly on confidentiality and/or availability-related conflicts. For example, security requirements ensure that the plant is shut down during an attack to prevent damage to the system. However, safety requirements ensure that the plant continues to run, provided that processes and/or process variables are within the desired limits.

ACKNOWLEDGMENT

The authors would like to thank the National Science Foundation Computer and Information Science and Engineering (CISE), award number 1846493 of the Secure and Trustworthy Cyberspace (SaTC) program: Formal TOols foR SafEty aNd Security of Industrial Control Systems (FORENSICS).

REFERENCES

- [1] E. A. Lee, "The past, present and future of cyber-physical systems: A focus on models," *Sensors*, vol. 15, no. 3, pp. 4837–4869, 2015.
- [2] S. Chockalingam, D. Hadžiosmanović, W. Pieters, A. Teixeira, and P. v. Gelder, "Integrated safety and security risk assessment methods: a survey of key characteristics and applications," in *International Conference on Critical Information Infrastructures Security*. Springer, 2016, pp. 50–62.
- [3] NSF. (2016) Cyber physical systems (cps) (nsf16549). [Online]. Available: <https://www.nsf.gov/pubs/2016/nsf16549/nsf16549.htm>
- [4] G. Kavallieratos, S. Katsikas, and V. Gkioulos, "Cybersecurity and safety co-engineering of cyberphysical systems—a comprehensive survey," *Future Internet*, vol. 12, no. 4, p. 65, 2020.
- [5] N. G. Leveson and J. L. Stolzy, "Safety analysis using petri nets," *IEEE Transactions on software engineering*, no. 3, pp. 386–397, 1987.
- [6] N. Leveson, M. Daouk, N. Dulac, and K. Marais, "Applying stamp in accident analysis," in *NASA Conference Publication*. NASA, 1998, 2003, pp. 177–198.
- [7] M. Burmester, E. Magkos, and V. Chrissikopoulos, "Modeling security in cyber-physical systems," *International journal of critical infrastructure protection*, vol. 5, no. 3-4, pp. 118–126, 2012.
- [8] C. Zhao, L. Dong, H. Li, and P. Wang, "Safety assessment of the reconfigurable integrated modular avionics based on stpa," *International Journal of Aerospace Engineering*, vol. 2021, 2021.
- [9] H. Wang, D. Zhong, and T. Zhao, "Avionics system failure analysis and verification based on model checking," *Engineering failure analysis*, vol. 105, pp. 373–385, 2019.
- [10] A. L. Dakwah and E. Villani, "System safety assessment based on stpa and model checking," *Safety science*, vol. 109, pp. 130–143, 2018.
- [11] G. Wu, J. Sun, and J. Chen, "A survey on the security of cyber-physical systems," *Control Theory and Technology*, vol. 14, no. 1, pp. 2–10, 2016.
- [12] R. Shrestha, H. Mehrpouyan, and D. Xu, "Model checking of security properties in industrial control systems (ics)," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 164–166. [Online]. Available: <https://doi.org/10.1145/3176258.3176949>
- [13] R. Neupane and H. Mehrpouyan, "An ontology-based framework for formal verification of safety and security properties of control logics," in *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2022, pp. 1–8.
- [14] S. O'Toole, C. Sewell, and H. Mehrpouyan, "Iot security and safety testing toolkits for water distribution systems," in *2021 8th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2021, pp. 1–8.
- [15] A. J. Kornecki and J. Zalewski, "Safety and security in industrial control," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, 2010, pp. 1–4.
- [16] M. Brunner, M. Huber, C. Sauerwein, and R. Breu, "Towards an integrated model for safety and security requirements of cyber-physical systems," in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2017, pp. 334–340.
- [17] R. Bramberger, H. Martin, B. Gallina, and C. Schmittner, "Co-engineering of safety and security life cycles for engineering of automotive systems," *ACM SIGAda Ada Letters*, vol. 39, no. 2, pp. 41–48, 2020.
- [18] A. Cimatti, R. DeLong, D. Marcantonio, and S. Tonetta, "Combining mils with contract-based design for safety and security requirements," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2014, pp. 264–276.
- [19] T. Hayakawa, R. Sasaki, H. Hayashi, Y. Takahashi, T. Kaneko, and T. Okubo, "Proposal and application of security/safety evaluation method for medical device system that includes iot," in *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, 2018, pp. 157–164.
- [20] S. Kriaa, M. Bouissou, and Y. Laarouchi, "A model based approach for scada safety and security joint modelling: S-cube," 2015.
- [21] F. Reichenbach, J. Endresen, M. M. Chowdhury, and J. Rossebø, "A pragmatic approach on combined safety and security risk analysis," in *2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops*. IEEE, 2012, pp. 239–244.
- [22] M. Gleirscher, N. Johnson, P. Karachristou, R. Calinescu, J. Law, and J. Clark, "Challenges in the safety-security co-assurance of collaborative industrial robots," in *The 21st Century Industrial Robot: When Tools Become Collaborators*. Springer, 2022, pp. 191–214.
- [23] C. Menon and S. Vidalis, "Towards the resolution of safety and security conflicts," in *2021 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2021, pp. 1–6.
- [24] T. Gu, M. Lu, and L. Li, "Extracting interdependent requirements and resolving conflicted requirements of safety and security for industrial control systems," in *2015 First International Conference on Reliability Systems Engineering (ICRSE)*. IEEE, 2015, pp. 1–8.
- [25] A. Al Farooq, E. Al-Shaer, T. Moyer, and K. Kant, "Iotc 2: A formal method approach for detecting conflicts in large scale iot systems," in *2019 IFIP/IEEE symposium on integrated network and service management (IM)*. IEEE, 2019, pp. 442–447.
- [26] M. Sun, S. Mohan, L. Sha, and C. Gunter, "Addressing safety and security contradictions in cyber-physical systems," in *Proceedings of the 1st Workshop on Future Directions in Cyber-Physical Systems Security (CPSSW'09)*. Citeseer, 2009.
- [27] S. Li, X. Cheng, X. Huang, S. A. Otaibi, and H. Wang, "Cooperative conflict detection and resolution and safety assessment for 6g enabled unmanned aerial vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [28] A. Kornienko, M. Polyanichko, and D. Efanov, "Methodological aspects of detection and resolution of conflicts of train control systems information security software," in *2016 IEEE East-West Design & Test Symposium (EWDTs)*. IEEE, 2016, pp. 1–5.
- [29] W. Yang, Z. Zhiyong, Y. Jianhua, and G. Lifen, "Static game approach for solving lane-merging conflict between autonomous vehicles," in *2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2016, pp. 53–57.
- [30] Y. J. Lu, B. Zhang, and Y. Li, "Conflict detection and avoidance strategy research of plane flight," in *Proceedings of the 32nd Chinese Control Conference*. IEEE, 2013, pp. 2387–2391.
- [31] W. Young and N. Leveson, "Systems thinking for safety and security," in *Proceedings of the 29th Annual Computer Security Applications Conference*, 2013, pp. 1–8.
- [32] J. Marques-Silva, I. Lynce, and S. Malik, "Conflict-driven clause learning sat solvers," in *Handbook of satisfiability*. IOS press, 2021, pp. 133–182.
- [33] J. Thomas, "Systems theoretic process analysis (stpa) tutorial," *Massachusetts Institute of Technology*. Retrieved from: <http://psas. scripts.mit.edu/home/wp-content/uploads/2014/03/Systems-Theoretic-Process-Analysis-STPA-v9-v2-san.pdf>, 2013.
- [34] G. Howard, M. Butler, J. Colley, and V. Sassone, "Formal analysis of safety and security requirements of critical systems supported by an extended stpa methodology," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2017, pp. 174–180.
- [35] N. G. Leveson and J. P. Thomas, "Stpa handbook," *Cambridge, MA, USA*, 2018.
- [36] J. P. Thomas IV, "Extending and automating a systems-theoretic hazard analysis for requirements generation and analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.
- [37] I. Friedberg, K. McLaughlin, P. Smith, D. Laverty, and S. Sezer, "Stpa-safesec: Safety and security analysis for cyber-physical systems," *Journal of information security and applications*, vol. 34, pp. 183–196, 2017.
- [38] W. Young and N. G. Leveson, "An integrated approach to safety and security based on systems theory," *Communications of the ACM*, vol. 57, no. 2, pp. 31–35, 2014.
- [39] S. Foucart and H. Rauhut, "An invitation to compressive sensing," in *A mathematical introduction to compressive sensing*. Springer, 2013, pp. 1–39.
- [40] J. P. Garrahan, "Simple bounds on fluctuations and uncertainty relations for first-passage times of counting observables," *Physical Review E*, vol. 95, no. 3, p. 032134, 2017.
- [41] J. K. Scott and P. I. Barton, "Bounds on the reachable sets of nonlinear control systems," *Automatica*, vol. 49, no. 1, pp. 93–100, 2013.
- [42] M. Fränzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert, "Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 1, no. 3-4, pp. 209–236, 2006.

- [43] G. Schollmeyer, “Computing simple bounds for regression estimates for linear regression with interval-valued covariates,” in *International Symposium on Imprecise Probability: Theories and Applications*. PMLR, 2021, pp. 273–279.
- [44] S. Zhang, M. Cao, and M. K. Camlibel, “Upper and lower bounds for controllable subspaces of networks of diffusively coupled agents,” *IEEE Transactions on Automatic control*, vol. 59, no. 3, pp. 745–750, 2013.
- [45] K. Scheibler, “Applying cdcl to verification and test: when laziness pays off,” Ph.D. dissertation, Dissertation, Albert-Ludwigs-Universität Freiburg, 2017, 2017.
- [46] G. Chen and T. J. McAvoy, “Predictive on-line monitoring of continuous processes,” *Journal of Process Control*, vol. 8, no. 5-6, pp. 409–420, 1998.
- [47] F. Capaci, E. Vanhatalo, M. Kulahci, and B. Bergquist, “The revised tennessee eastman process simulator as testbed for spc and doe methods,” *Quality Engineering*, vol. 31, no. 2, pp. 212–229, 2019.
- [48] M. Krotofil, A. A. Cárdenas, B. Manning, and J. Larsen, “Cps: Driving cyber-physical systems to unsafe operating conditions by timing dos attacks on sensor signals,” in *Proceedings of the 30th Annual Computer Security Applications Conference*, 2014, pp. 146–155.
- [49] J. Liu, W. Zhang, T. Ma, Z. Tang, Y. Xie, W. Gui, and J. P. Niyoyita, “Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection,” *Expert Systems with Applications*, vol. 158, p. 113578, 2020.
- [50] M. Segovia, J. Rubio-Hernan, A. R. Cavalli, and J. Garcia-Alfaro, “Cyber-resilience evaluation of cyber-physical systems,” in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2020, pp. 1–8.
- [51] C. Gao, H. Park, and A. Easwaran, “An anomaly detection framework for digital twin driven cyber-physical systems,” in *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*, 2021, pp. 44–54.
- [52] A. Ghafouri, Y. Vorobeychik, and X. Koutsoukos, “Adversarial regression for detecting attacks in cyber-physical systems,” *arXiv preprint arXiv:1804.11022*, 2018.
- [53] M. Krotofil, A. A. Cárdenas, B. Manning, and J. Larsen, “Cps: Driving cyber-physical systems to unsafe operating conditions by timing dos attacks on sensor signals,” in *Proceedings of the 30th Annual Computer Security Applications Conference*, 2014, pp. 146–155.
- [54] J. J. Downs and E. F. Vogel, “A plant-wide industrial process control problem,” *Computers & chemical engineering*, vol. 17, no. 3, pp. 245–255, 1993.
- [55] I. Kiss, P. Haller, and A. Bereş, “Denial of service attack detection in case of tennessee eastman challenge process,” *Procedia Technology*, vol. 19, pp. 835–841, 2015.
- [56] B. Genge, I. Kiss, and P. Haller, “A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures,” *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 3–17, 2015.
- [57] A. Bathelt, N. L. Ricker, and M. Jelali, “Revision of the tennessee eastman process model,” *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 309–314, 2015.