

Enhanced Cyber-Physical System Security: A Model Checking Approach Using UPPAAL for OT-Specific Applications

Dr Chetan Chauhan¹, Dr.M.Balaji², Azmeera Srinivas³, Abhay Chaturvedi⁴, Dr Shilpa P⁵, Mohit Tiwari⁶

Assistant professor, Department of Computer Engineering, Vishwakarma University, Pune, er.chouhan.chetan@gmail.com, Orchid ID: 0000-0001-7973-0644

Associate Professor, Department of Electronics and Communication Engineering, Mohan Babu University, Tirupati.balajichaitra3@gmail.com, Orchid id: 0000-0002-7693-581X

Assistant Professor, Department of ECE, Kakatiya Institute of Technology & Science, Warangal. srinivasazmeera85@gmail.com, Orcid id: 0000-0001-9957-2540

Associate Professor, Department of Electronics and Communication Engineering, GLA University, Mathura, abhaychat@gmail.com

Assistant Professor, Department of Mathematics, Dayananda Sagar Academy of Technology and Management, Bangalore, shilpap80@gmail.com

Assistant Professor, Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, A-4 Block, Rohtak Road, Paschim Vihar, Delhi, mohit.t.bvcoe@gmail.com

ARTICLE INFO

ABSTRACT

Received: 16 Dec 2024

Revised: 20 Feb 2025

Accepted: 28 Feb 2025

Introduction: Cyber-physical systems (CPS) unite computing elements and physical operations but remain vulnerable to cybersecurity threats which threaten human safety along with environmental safety. These systems experience high susceptibility to attack when cybercriminals conduct the five phases of communication while comprehending the IT and operational technology aspects of the targeted system. Security enhancement along with prevention of malicious activities requires immediate attention to these system vulnerabilities. This paper establishes an all-inclusive framework which includes specific recommendations that guide security assessment of CPS while focusing on operational technology (OT) and employing the UPPAAL model checking tool.

Objectives: A research method for CPS security verification serves as the main goal to develop security verification methods which meet inherent safety standards. A three-fold approach to CPS security management consists of establishing safety-oriented security constraints while building security threat detection systems and adding model checking protocols for defense system reinforcement. The research expands security modeling by designing a specific Attack Module which targets OT-based cyber threats. The research utilizes UPPAAL's dynamic simulation tools to identify precise security threats by performing system evaluations of human-computer interfaces while developing a continuous security assessment framework.

Method: The research handles security verification through structured utilization of UPPAAL which operates as a formal verification tool. The safety requirements of CPS OT components form the basis of security constraints at the starting point. A series of tests under model checking procedures ensures both the effectiveness and identification of potential vulnerabilities. Real-time reactivity and advanced simulation methods in UPPAAL allow users to properly define temporal characteristics of system behaviors. The OT-specific Attack Module permits operators to perform simulations of cyber attacks for determining system resilience through its integrated features.

Result: The requested method led to vital findings which verify its success in safeguarding CPS systems. The predefined security rules allow real-time detection of threats that originate from operating technology safety requirements. Model checking in cooperation with UPPAAL helps achieve exhaustive security measure verification. The Attack Module which was developed proved successful in creating OT-specific threat models which delivered important system vulnerability information.

Conclusion: Research findings demonstrate that formal verification remains crucial for protecting CPS against cyber attacks in actual systems. The researchers used UPPAAL simulation techniques to develop an effective framework that detects and resolves security weaknesses in Operational Technology systems. System protection improves significantly when security constraints receive structured implementation while an Attack Module is integrated into the

system. Security assessments under the proposed methodology run continuously to maintain long-term readiness against developing cyber threats in the environment. Research efforts in the coming period will concentrate on verifying the security evaluation procedures while adding support for different CPS frameworks to apply this method.

Keywords: CPS, formal verification, OT, security requirements, security assessment, UPPAAL.

INTRODUCTION

Thus, as the integration of the functionality of cyber and physical elements rises with the progress in CPSs, there lies the chance of revolutionizing innovations as well as the pinnacle of security threats. Present-day CPS applications are used in critical areas such as industrial control systems and self-driving cars, smart grids, and health care, etc. [1] such an integration demands robust methods for the availability of the system and security from all forms of maladies. However, out of all of these, model checking has been developed as one of the most efficient approaches of formal verification that systematically searches for the system states to meet the perimeters of security.[2] Consequently, model checking offers a method through which it is possible to determine whether CPS satisfies the specified security criteria while safeguarding them. This technique entails the creation of a rather formal model of the system and then getting through all the scenarios of the system and how they transition. [3]Furthermore, this approach assists in realization of insecure points and, thus, confirms that the system functions properly in various conditions. [4]To reinforce this approach even more, we used another feature of UPPAAL known as simulation that may search additional deeply into state spaces within several configurations of the models. This makes it possible to shift more consideration to the character of the user as well as study the relations between a human and the computer. Security verification was performed with the use of UPPAAL simulating tool; in the base of the system model, there was an Addition called Attack Module imitation of possible OT attacks. This model includes software and hardware subordinate, and the operator subordinate on the understanding that the intrusion of a CPS requires IT and OT professionals.

However, these issues are still relatively under discussed in more recent literature; most of the current works do not describe any distinguishable methodology for the generation of security constraints or incorporate OT-level attacks. This paper addresses this issue in two ways: first, for generating OT-level security constraints systematically; second, for using the model checking to verify the constraints. The key contributions of our approach are:

This research demonstrates that model checking offers a more comprehensive and efficient solution compared to traditional ad hoc testing and individual security solutions.[5]One of the approaches is in using a model to expose more subtle infirmities left undetected by common strategies, as the technique is not restricted to purely theoretical analyses demonstrating model checking's practical relevance in practical applications. For instance, in the case of control algorithms in autonomous vehicles, model checking guarantees that safety characteristics are met and that the car's behaviours are appropriate in new situations. In the same way, in industrial control systems, model checking confirms that the systems remain safe within the faults or cyber-attacks. The effectiveness of model checking also lies in its capability to provide formal guarantees of the systems' behaviour. If security properties have been stated in relatively unambiguous formalisms, one is able to gain a great deal of confidence in the system's secure operation by comparing actual operation against the model. This is a must in the scenarios where failure can mean a significant loss, for instance, in the case of life-sustaining structures like dams or health enhancing machines and equipment like heart pumps. That is why, model checking used together with other verification techniques — simulation and testing, for instance increases its efficiency. This approach offers an organised work plan that tends to accommodate for the theoretical and fundamental learning as well as the application of security to the system. To effectively defend the CPS technologies that are both emerging and are becoming indispensable on the cusp of the fourth industrial revolution, one has to harness the model checking techniques in order to secure and safely manage these systems against novel threats.

2. Background

2.1. Security in Cyber-Physical Systems (CPS)

The security factors related to CPS have been obsolete in recent years. Positive Technologies conducted a study that described the trends of vulnerabilities from 2017 to 2018. Table. 1 As for the key trends in 2017, the majority of vulnerabilities concerned the human-machine interface (HMI) and Supervisory Control and Data Acquisition

(SCADA) systems. These components are necessary for observing and managing industrial processes, which make them very attractive in the sight of the attackers. As of the year 2018, there was an increase in targeting Industrial control systems though HMIs and SCADA remain a critical issue. New threats suggest an increased variety in threats that may affect internal networks and specialized facilities. Bullying is more versatile, is used by attackers more often, and is based on the use of a wider range of classical and information technologies. Furthermore, a report that was published by TechRepublic in the year 2020 showed that of the 365 threats that were discovered in ICS systems, 75% were high and critical risks. This confirms the importance of higher protection levels of CPS against these increasing threats.

Table.1. Relevant comparisons between two approaches

Approach	Concept	Verification	Tool	Check
R. Akella	Information flow security	Information flow models provide reasoning	CoPS	Security process algebra
Our	Operation technology attacks, cyber-physical constraints	Modeling checking merits: system completeness and input coverage	UPPAAL	Temporal logic

2.2. UPPAAL Model Checking : Thus, to analyze the possible operational scenarios of a CPS and identify potential issues in their functioning, we utilized UPPAAL, a powerful model checking tool that would allow us to explore all the aspects of the problem. The features of the simulation are marvelous in the sense that UPPAAL is mostly used in conducting searches across different model configurations with high efficiency. This makes it possible to investigate the H/M dynamics, which is critical to the study of the CPS applications since the latter exhibit rather sophisticated behavior patterns. Derived by the Department of Information Technology in cooperation with Uppsala University and the Department of Computer Science at Aalborg University, UPPAAL is a tool for specific formal model checking for real time systems. It is derived from timed automata theory; thus, taking care of the requirements of systems where time is an essential factor. [6] There are still some areas in modeling which have not been well covered by most of the existing tools, however, UPPAAL's modeling language is quite flexible that it offers bounded integer variables with a specific query language to ensure system verification. Looking at the important aspects of the Mechanical check in the UPPAAL tool, there are several steps considered. First, the description of the arising system should be carried out, including the security specifications and design parameters. Fig. 1 Users then define such requirements in the model. The model checker of UPPAAL analyses if the given model comprehensively fulfils the outlined standard.[7] For any case that the model does not match the specifications, the tool comes up with counterexamples of where and how the discrepancies occur. UPPAAL offers three primary functionalities: The major component of the system comprises of the system editor, simulator and the system verifier. The system editor is used to implement the structure of the modules, which consist of components that describe the operational status and the transitions of the modules. Following the creation of the model, the simulator can be played out by step-by-step tracking or by random and portrays the status of the system. The verifier allows the users to introduce queries in temporal logic to know if the model satisfies certain conditions. It offers very specific feedback on whether the model adequately meets the needs or not; offers on the areas that may require further tweaking.

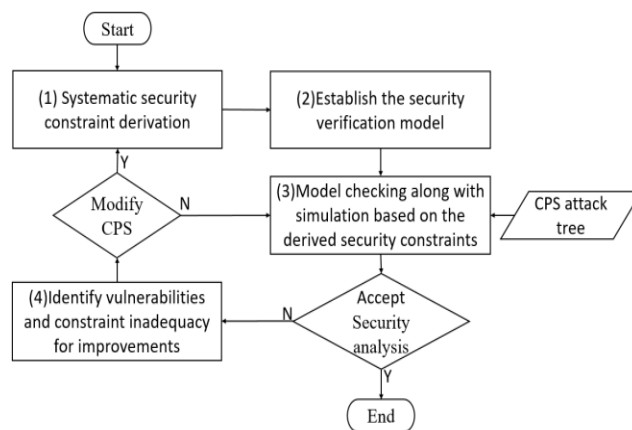


Fig.1. Proposed UPPAAL model.

OBJECTIVES

The area of CPS that our proposed framework for verifying security actually targets is carefully constructively designed to safely contain threats. This framework is divided into two primary phases: the generation of security constraints and an evaluation of the generated constraints with regard to their applicability with the process. First of all, these constraints are based on safety analysis, which provides the scope of needed security measures. These constraints form the premise of the next stage in which model checking is used to assess the constraints against possible security threats. In the first phase; that is the systematic generation of security constraints; the aim is to generate substantial security constraints that serve as the foundation of the verification.

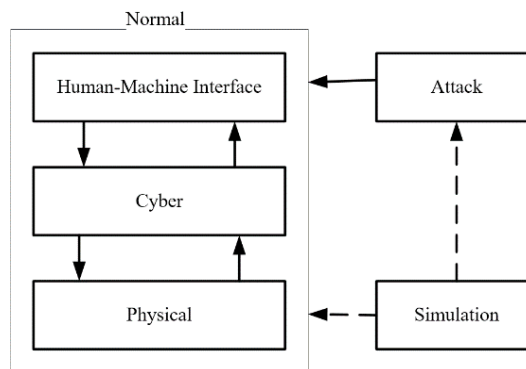


Fig.2 Process of the security verification approach.

This starts with a risk and safety assessment of the CPS, to determine the risks that might be present as well as the threats that may be posed. From this analysis we define the security constraints which are specifically targeted in order to address and neutralize these risks. Fig. 3 Each constraint is documented in an exhaustive manner in order to make each definition unequivocal and precise and thus build the basis for the verification process that follows. The second phase deals with security verification by checking set-up constraint with model checking. Model checking consists in generating a formal model of the CPS and exhaustively examining all the potentials states and transitions to verify that no of the constrained related to security are violated in any possible attack.

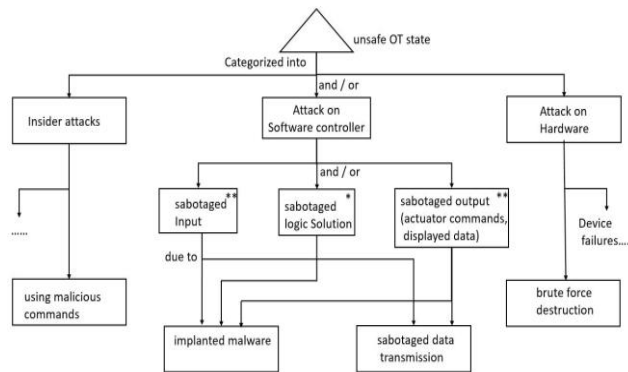


Fig.3 CPS attack tree templates.

This phase involves testing of various attacks to determine if the set constraints will help in mitigating them. On this basis the specificity of the constraints is improved to meet both security requirements and enhancing the system performance, and the necessary corrections are made where there were gaps or excessively stringent measures. This process goes on until this condition of providing adequate security for the CPS is met. [8] All in all, it can be evidenced that the approach presented here systematically evaluates cyber-physical systems against security threats. The two phase approach from safety analysis to detailed model checking along with constraint refinements do not only explore and counter threats but also improve the system's security state in the next iteration.

3. 1. Generating Security Constraints

In the fashion of security verification, we initiate by developing proscriptive security conditions that target possible OT attacks. This means creating value and range boundaries and checking that device states and process variables like water levels, pressures are within failure or hazard limits.[9] If causes and effects were dependency pairs, then dependency constraints mimic them and determine the possible actions a device can perform based on what it is expected of it and identify discrepancies. Cyber-physical liveness requirements ensure the correlation of the digital model to the physical one required to detect a possible cyber-attack or a faulty sensor. Global invariants impose basic constraints that keep the system running in an efficient and stable manner. During definition of the cyber model, restrictions are applied to make the logical variables of the nodes correspond to the real sensors and actuators. Likewise, the physical model guarantees that physical sensor values are well mapped into the cyber space. Fig.4 HMI design strategy assures that displays present information in the operator's interface that is best reflected the physical state to enable effective decision making. Global variables are controlled today in order to provide the necessary level of consistency and reliability of the results obtained both at the system level and at the level of individual components while performing their functions and meeting certain critical operational limits and safety conditions. This approach requires the overall integration of the security protocols, hence guaranteeing that the OT system is secure against cyber- physical attacks while enhancing the over-all security and reliability.

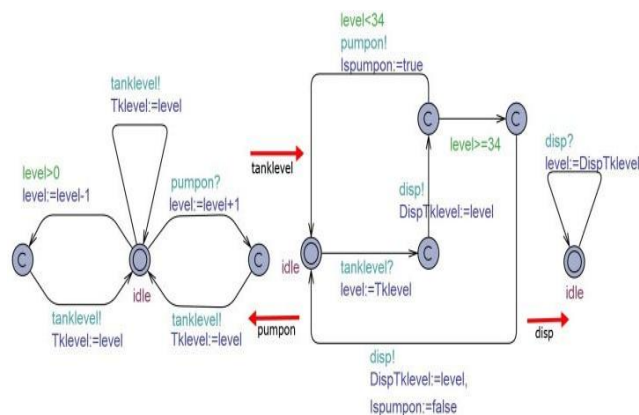


Fig.4 Communication and data exchange between UP AAL models.

3.2. Security Verification and Refinement

The security verification and refinement step kick off with model checking whereby verification queries are generated in accordance with the predefined security constraints and thereafter, the model checker is used to emulate different kinds of attacks. This entails employing ASM with the goal of modifying the behaviors of the attacked systems and consequently making them transit through various states that present different scenarios to evaluate the efficiency of the put constraints. However, more importantly, the analysis of the model checking results is necessary for exploring the possible loophole and weakness of the present protection solutions.[10] Analyzing these outcomes, it is possible to reveal vulnerabilities that can be exploited to penetrate the system. From these results, improvements are suggested to strengthen the system which might involve the addition of more sensors, the application of hardware precaution measures or improving the current constraints. This process of subsequent cycles guarantees that new threats and forces endangering the system, are eliminated, as well as the use of new kinds of attacks, constantly providing superior protection for the system.

3.3. Generating OT-Level Attack Scenarios

Subsequently in the process of creating OT-level attack scenarios, the initial step is the definition of objectives and the outcomes of an attack. This, in turn, involves pinpointing specific strike scenarios that would be disastrous to the system's overall function. For instance, an attack on a heat controller may be achieved in a manner that the controller fails to control heat and this results to fire outbreak. Fig.5 The expected outcome if such a situation were to occur would be that the controller is damaged, and in addition to not controlling temperature appropriately, it will feed operators with untruthful information, and they will not be able to make the necessary corrections on time. This textual outline of objectives and outcomes helps to make the scenario development more specific, which in turn enables the realistic sketching of attack models in terms of the consequences and effects on the changes in the physical and logical states of the system. Next, elaborating on the realistic attacks and their usage involves the steps, which are frequently carried out with the help of an attack tree model. This structured method is quite useful in visualizing and analyzing the potential paths an attacker could undertake in the achievement of their heinous plans. The mechanism of attack trees divides an attack on a system into its essentials, such as input attacks, output attacks, and logic tampering. [11] On the other hand, an output attack could involve manipulating the signals passed from the controller to the monitoring systems, this would cause wrong information from the displays to the operators. The last fundamental aspect of game is the logic tampering, which can be as simple as altering the software logic in a way that would confuse the operators and create circumstances that the design of the system does not cater for, thereby endangering the system. Based on the analysis of these scenarios via attack trees, it will be easier for the security analysts to understand the risks that the system faces and distinguish the best strategies to minimize the risks.

3.4. Constructing and Integrating Attack Modules

The first thing that needs to be done in the construction and integration of the Attack Modules is to attach Identifiers to all the described attack scenarios where each case can be easily identified and processed in the model. These identifiers enable setting up controls on the frequency and way each of the attack scenarios can be launched during simulations. The next step is to add the attack scenarios described to the normal system model or the roles, responsibilities and matrix model as applicable. This involves incorporating specific changes that signify possible attacks in the Attack Module to form a complete emulation infrastructure for normal operative status and that of malware attacks. For instance, if the attack aims to compromise sensor information, the model needs to be adapted to demonstrate how compromised information alters the systems' response. After these changes are implemented, the field of Simulation Module is set for controlling the procedure for running the attack scenarios. This includes defining conditions under which attacks take place, the manner and the expected results. The Simulation Module then analyzes the effects of these attacks on the system and gives insights and assessments on the System Administrator's protective measures. The proposed approach systematically develops security constraints, models different types of attacks, and utilizes model checking; thus, it promises to improve the security of CPS by several orders of magnitude. The methodology also guarantees that these systems are well protected against likely adversities since the approach involves massive simulation and assessment of the systems. It plays a role in constructing systems that can cope with a range of security issues in advance thus boosting their reliability and security. This cyclic practice not only strengthens existing capability against threats but also makes a coping mechanism for new and unexpected security threats, thus helping to implement a more secure CPS environment.

4. Illustrative Security Verification Approach

In the limited example of the automatic water injection system, we have considered above, the systematic derivation of the security constraints is critical to making the system dependable and defendable in face of various threats. It starts with the determination of value and range specifications that ensure important variables remain within specified limits. Indeed, the level of water in the tank must never be equal to or exceed 34 units. This constraint makes certain that the system is not filled beyond certain capacities, which may cause serious damages. Subsequently, dependency relations are set up in order to govern the proper coupling of various aspects of the system. In this case these constraints require that the controller should start the water injection process as soon as the tanks water level reduces to 34 units and should add water into the tank to make the level 35 units each time this process is initiated.

On the other hand, the controller must stop injection if the water level gets or is anticipated to be above this level. [12] This dependency constraints make the system always respond correctly to the changes in the water level, thus making its operation to be stable and predictable. Cyber-physical integrity is another important that has to do with security constraints. This entails ensuring that the data from the water level sensor relative to water level in the tank is correct and is correctly interfaced to the operator and acknowledged by controller. For instance, If the sensor records the water level to be 30, then this is the value that should be displayed and the one that the controller will use to make decisions on the injection of more water. It is of great importance to sustain this kind of consistency for the operator that will help him in his situational awareness and the system as whole.

Finally, global invariants guarantee that some important characteristics of the systems are preserved during the functioning and these characteristics can be, for example, energy or material balance. Regarding the above water injection system this can imply that the total volume of water injected must not raise the total level to more than 34 units. These invariants have to be sustained to enable the system to function without posing conditions that may cause it to fail or malfunction. All in all, based on such systematic generation of security constraints, the framework of a sound security verification strategy is developed. By taking care of value and range constraints, solving the dependencies' constraints, achieving cyber-physical consistency as well as the same global invariants, we can guarantee the proper and secure functioning of the automatic water injection system even if there are threats from the side of globalization, including cyber or physical ones.

4.1. Developing the Security Verification Model

In the development of a security verification model for our automatic water injection system, UPPAAL is employed to create a comprehensive and detailed simulation environment. This model is designed to test the predefined security constraints and ensure the system's robustness against potential threats. The process begins with the construction of a normal model that integrates the Human-Machine Interface (HMI), physical model, and cyber model, each representing crucial components of the system. Fig.6 The Cyber Model is represented by the Soft Controller, which is responsible for receiving data from the water level sensor and making decisions to control the water injection mechanism. This component is pivotal as it processes sensor data and initiates actions to maintain the water level within the specified range. The Physical Model is embodied by the Tsensor, which measures the actual water level in the tank and communicates this information to the Soft Controller. The accuracy and reliability of this sensor are critical for the system's proper functioning, as any discrepancy can lead to incorrect actions by the controller. Finally, the HMI Module is represented by the Udisplay which shows the current water level as determined by the Soft Controller.

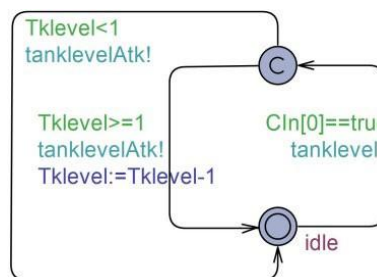


Fig.6 Restricting the attack to be activated in the malware input module.

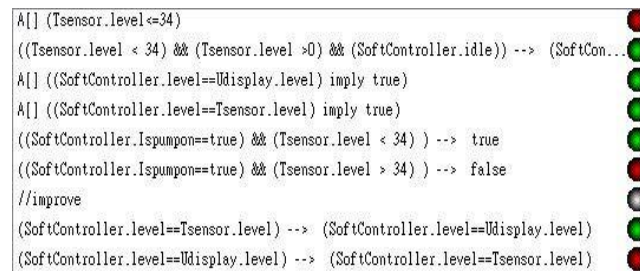
To perform a vigorous analysis of the system's vulnerability, several distinctive Attack Scenarios are constructed using the templates of the Cyber-Physical System (CPS) attack tree. There are two types of attacks modeled here: they mimic various actions to assess the system's ability to support and resist mischievous operations. For instance, the input/output transmission attack may entail changing the output the sensor gives to that of a lower water level than the real one. It simply could give the system a wrong signal to inject more water than required and thus leading to overflow and causality of the system. To integrate the mentioned attack scenarios, a new component called Malicious Attack Module is added to the UPPAAL model. [6] Among the many modules, SoftCtrlAtkIn is located between the SoftController and Tsensor wherein it receives data from the sensor and alters it. In this striking attack, the value of the 'water level' reading is decreased by one notch artificially. This manipulation checks the ability of the system to handle such false data input and the effects that it has on the system. With this attack module included in the verification model it is possible to now test how the system holds up to real attacks and make determination as to how strong the system is regarding its integrity and performance. As a result, normal control functions, as well as prospected attacks, can be examined in the operational structure derived from this security verification model built in UPPAAL. This way, it is guaranteed that the automatic water injection system that is to be developed is not only going to be designed to function efficiently in standard conditions but is also going to be protected against a variety of threatening agents. Such a methodical verification approach is crucial to reveal and counter threats, which in its

turn, contributes to the increase of the security level and stability of the system.

4.2. Model Checking and Simulation

To validate the system's ability to withstand attacks with high level of credibility, typical Attack Scenarios are developed using the structure of the Cyber-Physical System attack tree. These are very similar to the attack scenarios because the former involves various actions that may be performed by the attacker or some specific tool to assess the possibility to breach into the target system. For example, an input/output transmission attack could take form in modifying the output of the sensor so that it displays a lower water level than it holds. Such tricking can force the system into injecting more water than has been required, a condition that may force an overflow and hence destroy the system. For including these attack scenarios, the Malicious Attack Module is incorporated into the UPPAAL model. Of these, SoftCtrlAtkIn is interposed between the SoftController and Tsensor and modified the data coming from the sensor. In this type of attack, the water level reading indicates a fabricated low level by decrementing it by one. Fig.7 It is a type of input signal that examines the system's capacity to identify and neutralize fake data inputs. Thus, having included this attack module, the verification model will be able to emulate real attack scenarios and determine how well this system can preserve its key properties and qualities. In this way, the normal running components and different possible attacks can be included into one UPPAAL model of security verification and the real constraints of the system can be effectively examined. Thereby, it guarantees that the automatic water injection system not only will function appropriately when there are no threats but also will be protected from various malicious groups. [8] This methodical verification process helps to find the weaknesses and preconditions in the system and, thus, strengthen its security.

The situation of the system as seen through the current sensor and display should, therefore, be consistent with that controlled by the microcontroller. This avoids disparities that may result in the wrong calculations or questionable information to the operator. The queries $A[]$ (SoftController. Level == Tsensor. level) and $A[]$ (SoftController. Level == Udisplay. Level) check that the water level sensed by the Tsensor matches the level acknowledged by the SoftController and shown to the operator. This consistency is important to keep people's confidence in the readings of the instruments and to make precise adjustments.



```

A[] (Tsensor.level <= 34)
((Tsensor.level < 34) && (Tsensor.level > 0) && (SoftController.idle) --> (SoftCon...
A[] ((SoftController.level == Udisplay.level) imply true)
A[] ((SoftController.level == Tsensor.level) imply true)
((SoftController.Ispump == true) && (Tsensor.level < 34) ) --> true
((SoftController.Ispump == true) && (Tsensor.level > 34) ) --> false
//improve
(SoftController.level == Tsensor.level) --> (SoftController.level == Udisplay.level)
(SoftController.level == Udisplay.level) --> (SoftController.level == Tsensor.level)
  
```

Fig.7. UPPAAL system security constraint verification query

Global Invariants are responsible for the general consistency of the system's functioning. The first and one of the most important invariants is W, it is impossible to remain weak when the pump is on, but the water level must be less than 34 units. This is as confirmed by the queries such as (((SoftController. Ispump == true) && (Tsensor. level < 34)) -> true) and (((SoftController. Ispump == true) && (Tsensor. level > 34)) -> false). These queries ensure that the pump only runs when needed and any condition that will make the water level go beyond the recommended limit is prevented. This invariant is crucial to avoid overflowing and, thus, guarantee that the system's parameters are the ones initially set. By performing these queries in an orderly manner in UPPAAL, the model checking process thoroughly validates the system's adherence to the imposed security requirements. Such a verification assures that the automatic water injection system works in a correct way when the system is not threatened but also when the threats occur. Fig.8 The end-product is therefore a system that can work well to control water levels and protect the system against malicious attacks and any other related issues that may hamper its functionality hence improving on the general security and operational capacity. When the security checks are performed with the help of UPPAAL, the results will state if the system meets the constraints defined. If all such factors are properly managed and the system performs within these imposed limitations, it proves that the applied security measures are efficient. However, if anomalies or violations are detected such as water level going beyond the permissible limit or there are discrepancies between the sensor output and the display the output implies that there may be creeping in vulnerabilities in the security system. Such violations point out where there may be a need to go and research more on the actual system design or security limitations.

Identifying Vulnerabilities and Improving Constraints

The last of the steps in the security verification process is thus to review and strengthen the security constraints in

accordance with the results of the UPPAAL analysis. This involves several key activities,

Identifying Issues: This means that appearance of any of these constraints will indicate poor security posture in today's environment or a flawed Security Architecture. Fig.9

Constraint Enhancements: Besides, to solve the mentioned issues, it could be required to implement more security checks and constraints. New constraints that can be introduced are exemplified by the observation of significant differences in sensor readings with the controller data and the display values. This could encompass the inclusion of checks to validate that the water level reported by the sensor was accurate with the level defined by the controller and also the level displayed to the operator. These enhancements assure the system has proper and correct information throughout the system, which would eliminate uncalled actions and increase dependability.

```
A[] (ProcTprSensor.T==ProcHeater.T) imply true
ProcUser.UschOn==false --> ProcHeater.off
ProcUser.UschOn==true --> ProcHeater.heating || ProcHeater.idle
ProcHeater.T>200 --> HOn==false
ProcHeater.T<150 --> HOn==true
A[] ((HOn == false) && (ProcHeater.T > 200) imply true)
A[] ((HOn == true) && (ProcHeater.T > 200) imply false)
```

Fig.8 Results of security constraint detection.

```
((SFCntr.IdleOutput && AlarmOn==false) || (SFCntr.T1Output && AlarmOn==false) || (SFCntr.T2Output && ...
A[] (T1.PumpOn==true && T1.ValveOn==true) imply (T1.Volum<= T1.LastVolum)
A[] (T1.PumpOn==false && T1.ValveOn==false) imply (T1.Volum<= T1.LastVolum)
A[] (T2.PumpOn==true && T2.ValveOn==true) imply (T2.Volum<= T2.LastVolum)
A[] (T2.PumpOn==false && T2.ValveOn==false) imply (T2.Volum<= T2.LastVolum)
(T2.PumpOn==true && T2.ValveOn==true) --> (T2.Volum<= T2.LastVolum)
(T2.PumpOn==false && T2.ValveOn==false) --> (T2.Volum<= T2.LastVolum)
A[] HMI.Display imply (T0.Volum==SFCntr.TOV) && (T0.Volum==HMI.iTOVDisp)
A[] HMI.Display imply (T1.Volum+T1.iSteamV==SFCntr.T1V) && (T1.Volum+T1.iSteamV==HMI.iT1VDisp)
A[] HMI.Display imply (T2.Volum+T2.LeakV==SFCntr.T2V) && (T2.Volum+T2.LeakV==HMI.iT2VDisp)
(T0.React && T1.React && T2.React) --> (T0.Volum+T0.T1FeedV+T0.T2FeedV+T1.Volum+T1.iSteamV+T2.Volum+...
(SFCntr.IdleOutput || SFCntr.T1Output || SFCntr.T2Output) --> (SFCntr.TOP+SFCntr.T1P+SFCntr.T2V) ==110
(HMI.Display) --> (HMI.iTOVDisp+HMI.iT1VDisp+HMI.iT2VDisp)==110
```

Fig.9 Verification results of system security constraints in the attack scenario.

Refinement: The enhancement of the security constraint and the design of the system entails the following. Overall, the fully developed system needs to be reiterated to ensure that the new constraints are truly useful in addition to the first levels of improvement. This is done through an organism like an increase or a decrease of the constraint depending on the results of the test carried out. This fundamental level must be reached so that all dangers are kept to a minimum and the computer system works properly every time a respective theoretical situation is realized. This cycle of testing, to observe and define some problems, improve constraints, and optimize the system design guarantees the effectiveness of the security measures. Not only have the present-day threats detected and neutralized, but the system becomes immune to future attacks as well due to this systematic approach. While checking the system with various types of attacks and enhancing the security constraints, the reliable framework is created. This framework protects the CPS from different types of attacks and enables the system to develop an effective countermeasure against those attacks, thus increasing the level of CPS security and destructiveness.

6. Case Study Analysis

To demonstrate the effectiveness of our proposed approach, we applied it to two different systems: The complex under consideration included a heating control system, and a safety injection system. The following section gives a focus on each case study to demonstrate how the uses of the methodology as well in analyzing and rectifying the security risks. The heating control system oversees a heater that is operated by a switch and the present temperature is shown on a display. The operational workflow is straightforward: the heating switch controls the working of the heater and at the same time is a display of the current temperature of the heater. In the same manner turning the knob off cools the system and on warms the system respectively. The heating is initiated if the temperature falls to below 150 degrees centigrade and increases the temperature up to the peak of 200 degrees centigrade. Several security constraints were defined in order to make the system work properly as well as to prevent various types of attacks. General constraints of value and range were set to adjust the temperature, they made heating safe and did not allow it to fall out a rage of 150°C to 200°C when heating is on. Additional states of dependencies were established for the purpose of putting the system into a cooling state when the switch is off and to a heating state when the switch is on as a reaction to the user's command. To eliminate cases where there is false data injected into the system, cyber and physical consistencies were incorporated, and these referred to the temperature as reported by the sensor as well as the temperature controlled by system software and as displayed on the user interface. Specific inputs had to be set to 200 degrees Celsius for global invariants to accomplish the regulation of the temperature and a proper ratio between the heat produced and the temperature that increases were to be maintained for the system stability.

Thus, with the help of UPPAAL, the model of the heating control system was designed, which also incorporates the HMI model, which remains responsible for control panel with switches and displaying temperature to properly represent the actions of the user within the context of system functioning. The cyber model specifies the heating control system's logic; it makes decisions on temperature control and heater management depending on the data from the sensors and users' commands. The physical model emulates the temperature sensor that measures the actual temperature and is vital for the system's operational decisions. To assess the vulnerability of the system two types of attacking strategies were analyzed. In CaseL," the malware interferes with the heater by making it stop working at 150oC and display that the temperature is increasing even though it is not, thus endangering the conditions and deceiving the operator. Case is like CaseL: however, different types of control and display manipulations are used to check the operational readiness of the system in response to various kinds of attacks.

With the combined help of UPPAAL we assessed the security requirements and the level that the built system constrains it to. Regarding the value and the range, we made sure that heating initiates at 150 degrees Celsius and terminates at 200 degrees Celsius to avoid reaching dangerous operational temperatures. Regarding the state dependencies, we confirmed that when the switch in the system is in the cooling state, the system provides cooling; when the switch is in the heating state, the system provides heating. To ensure cyber and physical integration, crosschecked whether temperature posted online is the same with that of the sensors for real, accurate and reliable information to be posted. Additionally, for global invariants it was verified that the temperatures never exceed 200 degrees, thus making the system safe and secure. This revealed that the interdependent constraints were not effective in identifying highly advanced threats and hence required changes to the constraints to optimize the level of identification.

The safety injection system deals with three tanks that are used in steam production and water handling. Tanko pumps in steam and returns the condensate water through Tank1. Water is pumped from Tank1 to Tanko and Tank2 is responsible for aggregating any spillage from Tanko. The system is intended to keep water tanks always full to avoid water tank dry-out, which is a safety issue and also offers alarms when full. We set up conditions to help achieve the best systematic workflow. The variables that were used included value with range constraints that were set mainly aimed at regulating the water levels in Tanko within L1 and L8. 5 with alarms attached that go off when the water level is not within this range to guarantee safe and appropriate levels of water. State dependencies were developed to make the system start injecting water in either Tank1 or Tank2 dependent on the water level in Tanko to keep running and not dry out. Measures such as cyber and physical consistencies were introduced to ensure real-time, sound and clean traced data, by matching the displayed values of water levels to real values. It was global practice to set invariants as a measure that would guarantee that the injection of water was smooth but be slightly above maximum levels to preserve the system.

RESULTS

In the attack scenario, the malware interferes with the normal functioning of the system and disables the correct water injection while altering the display figures that are fed to the operator causing a possible dry-out situation. In this paper, we employ UPPAAL to verify a system's security constraints that is infected by malware. The constraints tested were water level constraints, state transition constraints, and the constraints concerning checks. According to the results, the initial constraints became a problem because advanced malware could easily trick the system into thinking it is behaving correctly and displaying proper responses. To mitigate these issues, further restrictions have been suggested, for example, components with displayed values used in orders, which would increase the ability to identify contradictions. Applying the proposed method in such case studies revealed that there exist shortcomings of initial security restrictions and therefore the need for their advancement. The use of the given method also described inefficiencies and future amendments like the inclusion of consistency checks and redesigning the system to increase security. This systematic approach makes sure that the security constraints are every tested and improved thus serving as a secure platform for operation of the CPS. Therefore, the case study of heating control and the safety injection systems will affirm the suitability of the suggested security verification method. As seen in the approach of generating security constraints, developing a model checking and fine-tuning the constraints to check a particular verification model, this method provides solidity in identifying and eradicating influences that can make a system insecure. In the case of heating control system, the feasibility study revealed that rigid security measures fail in identifying a number of intricacies in an attack. In the detailed formulation of the constraints the actual critical shortcomings, for instance the risk of the system overheating and providing the consumers with wrong temperature data was identified. ; Expanding the list of constraints in iteration manner improving these constraints to include consistencies enhanced the navigational and identification capabilities of threats. This makes it clear that in systems with complex features of cyber-interaction with the physical world, a constant, highly detailed approach to checking the security of the identified vulnerabilities is necessary. The following is the application of the above stated methodology in relation to the safety injection system case study; The above-mentioned case study defined how the methodology contributed to the identification of the safety issues. To this end, further restrictions related to water levels, states of the system, and cyber-physical harmony were given as tests of the system's robustness Simulated

attack based on malware were also optimally carried out. It was found that initial constraints were not adequate to overcome manipulations but if extra constraints are incorporated, that try to match the physical parameters with the displayed values, this helps the system in identifying the anomalies and failures. Altogether, these case studies corroborate that the proposed methodology provides a structured framework for the improvement of CPS security.

CONCLUSION

The main goal of the project is shown to be achieved, depicting the application of the proposed approach to protect cyber-physical systems. The methodology of systematically producing, validating, and optimizing security constraints with help of elaborated verification models and abounded model checking, gives rather specific logical flow of how security problems should be solved. The holistic heating control system case established the disadvantage of the early constraint and demonstrated what is possible through iterative improvement. Such enhancements as consistent check also raised the issue of demand for growth-oriented security validation methodologies that would enable the continuous updating of a system's capability to respond to highly complex attack while guaranteeing safe functioning. Likewise, in the safety injection system case, the approach demonstrated the scope of the 'safety' issues that the system can solve. The addition to introducing constraints about water level and system states and fine tuning of these constraints based on the simulations about various attacks enhanced the robustness and preciseness of the system. Therefore, the project shows the need to have a comprehensive and flexible security verification process for CPS. By adopting detailed formulation of constraints and using model checking, along with iteration employing the idea of design refinement, the given methodology guarantees that the systems are fault-tolerant, not being vulnerable to numerous types of attacks. Thus, it not only strengthens the protection of individual systems but also lays a solid basis for future improvements in CPS security.

REFERENCES

- [1] CyberX. (2019). 2019 Global ICS & IIoT Risk Report. Accessed: Feb. 2, 2021. [Online]. Available: <https://cyberx-labs.com/resources/risk-report-2019/#download-form>
- [2] 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. 213, Jan. 2018, doi: 10.1109/TDSC.2015.2509994.
- [3] K. Zetter. (2016). Inside the Cunnning, Unprecedented Hack of Ukraines Power Grid, Wired. Accessed: Feb. 2, 2021. [Online]. Available: <https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/>
- [4] R. Akella, Veri cation of information ow security in cyber-physical systems, Ph.D. dissertation, Dept. Comput. Sci., Missouri Univ. Sci. Technol., Rolla, MO, USA, 2013.
- [5] Positive Technologies. ICS Vulnerabilities: 2018 in Review. Accessed: Feb. 2, 2021. [Online]. Available: <https://www.ptsecurity.com/ww-en/analytics/ics-vulnerabilities-2019/>
- [6] TechRepublic. Industrial Control System Cybersecurity Vulnerabilities are Rising in 2020. Accessed: Feb. 2, 2021. [Online]. Available: <https://www.ptsecurity.com/ww-en/analytics/ics-vulnerabilities-2019>
- [7] History. Blackout Hits Northeast United States. Accessed: Feb. 2, 2021. [Online]. Available: <https://www.history.com/this-day-in-history/blackout-hits-northeast-united-states>
- [8] Taiwan Transportation Safety Board. Releases Final Report on TransAsia Airways Flight GE 235 Occurrence Investigation. Accessed: Feb. 2, 2021. [Online]. Available: <https://www.ttsb.gov.tw/english/16051/16113/16114/16531/post>
- [9] AviationAccidents.ChinaAirlines-Airbus-A310-B4-622R(B-1814)Flight CI676. Accessed: Jan. 7, 2021. [Online]. Available: <https://www.aviationaccidents.net/china-airlines-airbus-a310-b4-622r-b-1814-ight-ci676/>
- [10] P. Martins, A. B. Reis, P. Salvador, and S. Sargento, Physical layer anomaly detection mechanisms in IoT networks, in Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS), Budapest, Hungary, Apr. 2020, pp. 19, doi: 10.1109/NOMS47738.2020.9110323.
- [11] D. Ding, Q.-L. Han, X. Ge, and J. Wang, Secure state estimation and control of cyber-physical systems: A survey, IEEE Trans. Syst., Man, Cybern. Syst., vol. 51, no. 1, pp. 176190, Jan. 2021, doi: 10.1109/TSMC.2020.3041121.
- [12] Cyber-Physical Systems Virtual Organization. Veri cation Tools Main Wiki Page. Accessed: Mar. 21, 2021. [Online]. Available: <https://cpsvo.org/node/32762>.