Introducing Cyber Security at the Design Stage of Public Infrastructures: A Procedure and Case Study

Sridhar Adepu and Aditya Mathur

Abstract Existing methodologies for the design of complex public infrastructure are effective in creating efficient systems such as for water treatment, electric power grid, and transportation. While such methodologies and the associated design tools account for potential component and subsystem failures, they generally ignore the cyber threats; such threats are now real. This paper presents a step towards a methodology that incorporates cyber security at an early stage in the design of complex systems. A novel graph theoretic mechanism, named Dynamic State Condition Graph, is proposed to capture the relationships among sensors and actuators in a cyber physical system and the functions that are affected when the state of an actuator changes. Through a case study on a modern and realistic testbed, it is shown that introducing security at an early stage will likely impact the design of the control software; it may also lead to additional hardware and/or software requirements, e.g., sensors, or secure control algorithms. Such impact on the system design promises to improve the resilience of a system to cyber attacks.

Keywords Cyber attacks · Cyber security · Cyber physical systems · Security by design · Dynamic state condition graph · SCADA · Water treatment

This work was supported by research grant 9013102373 from the Ministry of Defense and this work was supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-40) and administered by the National Cybersecurity R&D Directorate.

S. Adepu (✉) · A. Mathur (✉)
iTrust, Center for Cyber Security Research, Singapore University of Technology and Design, Singapore, Singapore
e-mail: sridhar_adepu@sutd.edu.sg

A. Mathur
e-mail: aditya_mathur@sutd.edu.sg

M.-A. Cardin et al. (eds.), *Complex Systems Design & Management Asia*,
Advances in Intelligent Systems and Computing 426,
DOI 10.1007/978-3-319-29643-2_6

Acronyms

CPS	Cyber physical system
DPIT	Differential pressure indicator and transmitter
DPSH	Differential pressure switch
LIT	Level indicator and transmitter
MV	Motorized valve
PLC	Programmable logic controller
PSH	Pressure switch
RO	Reverse osmosis unit
SCADA	Supervisory control and data acquisition
DSCG	State condition graph
SWaT	Secure water treatment
UF	Ultrafiltration unit
UV	Ultraviolet (dechlorinator)

1 Introduction

Cyber Physical Systems: A Cyber Physical System (CPS) consists of a physical process controlled by a computation and communications infrastructure. Typically, a CPS will have several Programmable Logic Controllers (PLCs) each with control software for computing control actions. Each PLC controls a portion of the entire process. The control actions are based on the current state of the system obtained through a network of sensors. When effected, and assuming the hardware effected is in working condition, the control action causes a desired change in the process state. For example, in a water treatment system, a PLC may start a pump to fill a tank with water to be sent through an ultrafiltration system. The pump must be stopped when the tank reaches a predetermined high level. The level of water in the tank is known to the PLC through a level sensor.

The PLCs in a CPS can be viewed as a system that transforms the state of the process as in Fig. 1. At any instant the PLCs receive data from sensors, compute control actions, and apply these actions to specific devices. Note that there are several potential attack points in a CPS. In this work only the communication links between sensors to PLC, denoted as a black blob in Fig. 1, are considered.

Response of a CPS under cyber attacks: The communications infrastructure of a CPS is often connected to an external network. Such connections render a CPS susceptible to cyber attacks. The presence of wireless communications among the CPS infrastructure, makes it even more vulnerable to cyber attacks. Such attacks could compromise the communications links between sensors and PLCs and among the PLCs. Once any communications link has been compromised, an attacker could use one of several strategies to send fake state data to one or more PLCs. Unless the defense mechanism of the attacked CPS is highly robust, such attacks could cause

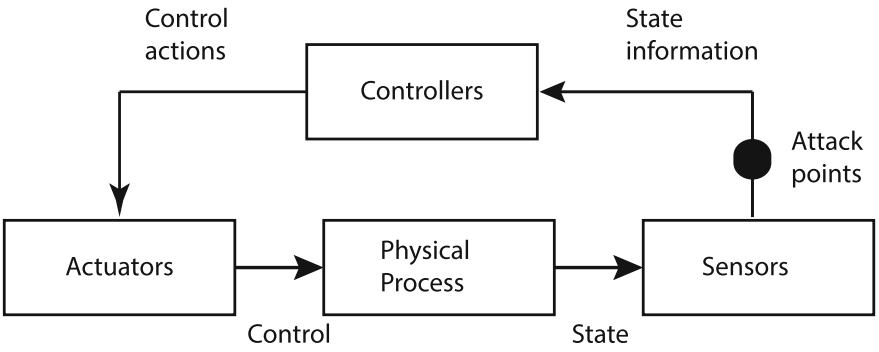

Fig. 1 CPS as a state transformer. In a water treatment system, actuators include pumps and motorised valves, while the sensors include level sensors, pH meters, chlorine sensors, and ORP (Oxidation Reduction Potential) meters. The *dark blob* indicates attack points considered in this work

an undesirable response from the CPS that may lead to system shutdown and/or device damage. Examples of such strategic attacks are given in Sect. 4.3. Thus, it becomes imperative for engineers to understand the response of a CPS to a variety of strategic cyber attacks and assess the robustness of its defense mechanism. An investigation like the one briefly described here is also critical in identifying errors in the control algorithms used by the PLCs though detection of such errors is not considered in this work.

Problem setting: It is assumed that a design consisting of various CPS components, and their interconnections, is available. For example, for a water treatment plant used in a case study, the physical subsystem of such a design would consist of pumps, tanks, valves, sensors, chemicals, etc., and the connecting pipes. The cyber component would consist of communications networks and various computing devices such as PLCs and other computing infrastructure. At this stage only the design of the physical system is available, and the control algorithms have not yet been coded. Prior to the actual construction of the CPS, and coding the control algorithms, it is desirable to know how would the system respond to cyber attacks. Thus, the problem can be stated briefly as follows.

(a) Using its design, determine how would a CPS respond to a set of potential cyber attacks and (b) how would the responses so obtained affect the design of the physical system and the control algorithms so as to improve its resilience to cyber attacks?

Contributions: (a) A scalable and automatable security-by-design procedure to understand the response of a Cyber Physical System to attacks on its communications infrastructure. (b) Dynamic State Condition Graph (D-SCG) as a formal modeling device for sensor-actuator constraints in a CPS.

Organization: The remainder of this work is organised as follows. Section 3 presents a step by step process for security by design of CPS. Section 4 presents a general CPS architecture, attacker models, and the DSCG. This section also

contains examples to illustrate a procedure based on DSCGs for impact analysis of cyber attacks. Section 5 presents a case study to demonstrate how an DSCG-based procedure can be applied to analyze the defense mechanism of an operational water treatment system. Questions regarding the novelty, automation, and scalability of the proposed approach are discussed in Sect. 6. Related research and how it differs from that presented here is in Sect. 2. A summary, discussion, and next steps in this research appear in Sect. 7.

2 Related Work

A large body of work focusing on the modeling and analysis of CPS systems is available. Given that this work is concerned with using special kind of graphs to construct a formal design procedure, in this section works related to graphs in CPS is considered.

Topological vulnerability analysis: Jajodia et al. [1] proposed a detailed procedure for modeling cyber systems using attack graphs. Such graphs model practical vulnerabilities in distributed networked systems. While attack graphs model vulnerabilities, DSCGs do not. In fact DSCGs simply model conditions required to control a component in a CPS; vulnerabilities, if any, are discovered through an analysis procedure described in this paper. The attack graphs and DSCGs are similar in the sense that both model all paths through a system. Note that while DSCGs are specifically designed to model CPS, attack graphs are not.

Control flow integrity: Abadi et al. [2] propose a control-flow integrity (CFI) approach to mitigating cyber attacks on software. They claim that CFI "…can prevent such attacks from arbitrarily controlling program behavior." CFI differs from DSCGS in many ways. First, CFI approach targets only software whereas DSCG target both hardware and software. Second, CFI is aimed at ensuring that malware does not affect the behavior of software. DSCGs do not focus on malware. Third, as mentioned above, DSCGs aim at modeling the behavior of controllers and hence obtain only conditions that must be true for a control action which is not the focus of CFI.

Other modeling approaches: Chen et al. [3] have proposed argument graphs as a means to capture the workflow in a CPS. The graphs are intended to assess a system in the presence of an attacker. The graphs are formed based on information in the workflow such as use case or state, physical system topology such as network type, and attacker model such as order to interrupt, power supply, physical tampering, network connection, Denial of service, etc. Vertices in these graphs contain the information corresponding to certain classes of security related information; they do not capture conditions for successful control actions. Instead, the graphs assume that the existence of flow implies a secure system.

Argument graphs are considered corresponding to each use case. When all use cases are considered, the graphs become unwieldy and difficult to analyze. The DSCG graphs are drawn from the conditions used in a CPS or its design, and

hence enable the analyzes of the damage or other effects when properties of the system are altered as may happen when an attacker enters a system via cyber or physical means.

Would an extension of argument graphs make them look and function like DSCGs? The answer is in the negative because the inputs to argument graphs and those in DSCGs are different. The vertices in DSCGs are physical devices and edges are conditions to initiate actions. In argument graphs the vertices may contain information about sub-steps in the use case, attacks and network related devices. Attack graphs are mostly helpful to find out the security level of the system in "Layer 1" and "Layer 2" in the SWaT system whereas DSCGs are generated based on the level 0 physical devices and their control dependency on one another.

Typed graphs [4] and Baysian defense graphs: [5] are a few other important contributions to the modeling of cyber attacks. However, once again, these differ from DSCGs in aspects already mentioned above.

Robust CPS: Here exists literature on the design of robust CPS [6–8]. These works focus on attack modeling, the design of controllers and monitors for secure CPS. In this paper, attack models are borrowed from Cardenas' work [9] as this is most closely related to cyber attacks in a CPS. Many other works model attacks specifically on control systems and are abstract in nature. DSCGs allow one to model attacks in a very practically visible way through the use of design diagrams.

Robust CPS: There is a large body of literature on failure mode and effects analysis of physical systems [10, 11]. These methods, also referred to as FMEA, are aimed at assessing the reliability of a system in the presence of failure of its components. Indeed, FMEA is a useful technique for assessing the reliability of the physical portion of a CPS. However, the presence of control software renders FMEA difficult and sometimes impractical to use due to software complexity. In fact the analysis procedure presented here using DSCGs can be considered as a new variant of FMEA and more appropriately referred to as SMEA: Security Mode Effects and Analysis. Note that, though not discussed here, failure can also be modelled using DSCGs via minor modifications to the graph semantics.

3 Security by Design: A Process

CPS design begins with a clear understanding of the requirements. These requirements indicate the expected, behavior of the CPS to realise an end objective such as deliver electric power to customers or produce clean water. How failure of system components needs to be managed is either a part of the requirements or added using techniques such as failure mode and effects analysis. Experience with the design of CPS suggests that while component failure scenarios are included in the CPS design, cyber attack scenarios are not. While some cyber attacks might lead to failure-like conditions, other strategic attacks might not. Even though a cyber-attack might lead to a failure-like scenario, the malicious intent of an attacker significantly increases the occurrence probability of an otherwise low-probability

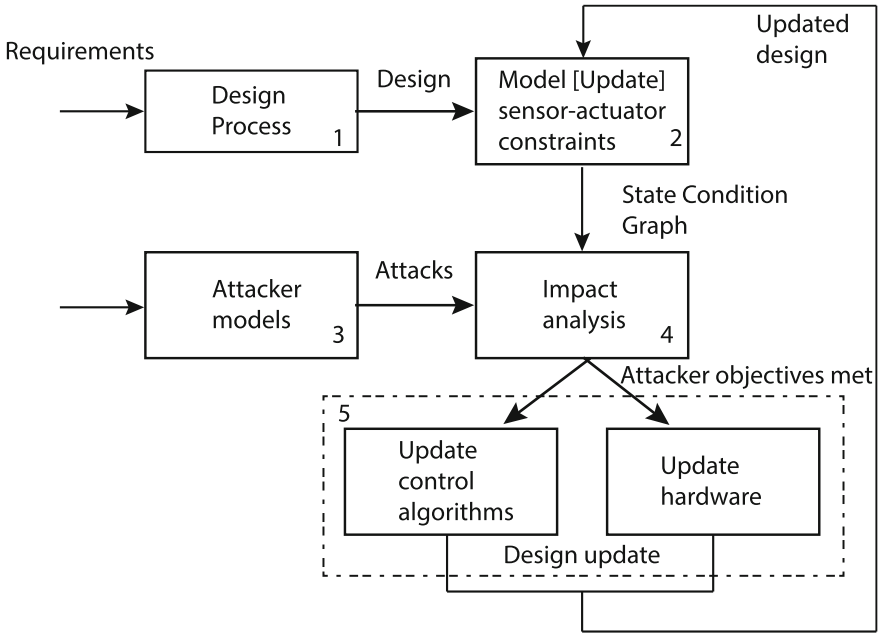

Fig. 2 An iterative design process to strengthen hardware and control algorithms to mitigate the effect of cyber attacks

failure scenario. This makes it important to include a security analysis component in the design process described below with reference to Fig. 2.

Step 1: CPS design: Given the CPS requirements, a design team creates the design of a physical process. This design is in the form of an engineering schematic, supported by textual explanation such as in [12]. The schematic shows the various physical components, their interconnections, and possibly mechanical and electrical specifications. It is assumed here that the computer programs for controlling the CPS are written after at least an initial physical design is available. Certainly, in some cases, code could be borrowed from a previous similar design.

Step 2: CPS model: The design of the physical system is then used to derive constraints that must be satisfied for each actuator in the system to be in a given state. Grouped together, these constraints lead to an DSCG as explained in Sect. 4.2.

Step 3: Attacker model: An attacker is modeled in this step in terms of objectives and the attack means. As explained in Sect. 4.3, each attacker model may lead to more than one attack.

Step 4: Impact analysis: As described in Sect. 4.4, attacks generated using attacker models are used as input to the DSCG. The ensuing analysis leads to the response of the CPS to different attacks and whether or not the attacker objectives are met. A given attacker model may generate an unusually large number of attacks. Thus, the choice of which attacks to select for impact analysis becomes an issue.

Later we explain how a DSCG aids in making such a choice based on economic arguments.

Step 5: **Design update**: Impact analysis reveals whether the attacker objective can be met or not. The impact analysis could be carried out with or without the PLC control software in place. If the PLC software is in the model, the impact analyses reveals any weaknesses of the software, such as a missing check that might otherwise reveal a cyber attack or a component failure. An indication that the attacker objective can be met implies weaknesses in system defense assuming the current design. It also offers clues as to what hardware and software defense are needed to reduce the chances of the attack being successful. Any changes made in this step in hardware design requires updates to the model (Step 2) and re-execution of steps 4 and 5. The loop consisting of steps 2, 4 and 5, ought to be executed in the design process as long as the impact analysis reveals weaknesses in the design based on the attacker models. This process is illustrated in Sect. 4.5.

4 Modeling a CPS

The first step in the proposed procedure is to construct a suitable model of a CPS. While a model using tools such as Simulink [13] or Labview [14] are certainly aids in the procedure described here (especially in Step 4 in Fig. 2), the procedure described here is a complementary aid. A general architecture of a CPS and the modeling procedure based on this architecture, are described next.

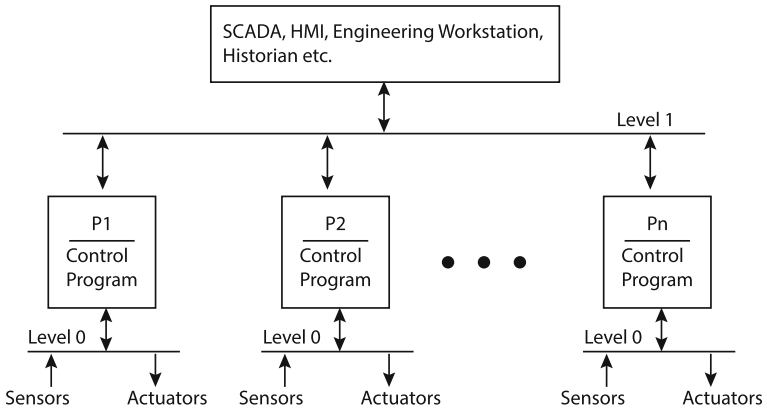

Fig. 3 Architecture of the control portion of a CPS. P1, P2, …, Pn denote PLCs. Each PLC communicates with its sensors and actuators through a local network at Level 0. PLCs communicate among themselves via another network at Level 1. Communication with SCADA and other computers is via a Level 3 network not shown here. Note that the actuators, e.g., a pump, also have sensors to indicate their condition

4.1 Structure of a CPS

CPS, such as power grid and water treatment systems, consist of a distributed supervisory control system. The control system itself is a collection of PLCs each controlling a specific portion of the CPS. This architecture is exhibited in Fig. 3. As shown, each PLC communicates with a set of sensors and actuators via a local network. This network is considered to be at Level 0 and is also referred to as the field-bus network [15]. The PLCs communicate with each other using the Level 1 network. Such a layered network structure is in accordance with the prevailing practice for industrial control systems [16].

As in Fig. 3, each PLC is responsible for the control of a set of actuators. The control actions are computed based on data received from a set of sensors local to it as well as data obtained from other PLCs. Data from sensors local to other PLCs is obtained via the Level 1 network through a sequence of message request and response.

Each PLC contains a control program that receives data, computes control actions and applies these to the actuators it controls. Computation of control actions is based on a condition evaluated using data received from the sensors. This could be a simple condition involving data from one sensor, or a compound condition involving data from multiple sensors some of which might be communicating with other PLCs. As described next, it is these conditions that are captured in the form of an annotated dependency graph to create a model for a CPS.

4.2 Dynamic State Condition Graphs

A Dynamic State Condition Graph (DSCG), is a pair (N, E), where N is a finite set of labeled nodes and E a finite set of (possibly) labeled directed edges. Three types of nodes are considered. A state-node, referred to as *s-node*, denotes the state of an actuator such as a pump, a tank, a generator, or a tap changing transformer. For an actuator with k states, there are k nodes in the corresponding DSCG, one denoting each state. A component node, referred to as a *c-node*, denotes any component of a CPS that could be in any of two or more states. An operator-node, referred to as *o-node*, denotes a logical operator such as a logical and ($\wedge$), logical or ($\vee$), and logical not ($\neg$).

A labeled edge is a triple (n_1, l, n_2), where n_1 and n_2 denote, respectively, c-node and s-node, and l the state of the component denoted by n_1. n_2 denotes the state of an actuator. l could be specified as a condition, such as *pressure less than 3 Bar*, or as a discrete state, such as `C(losed)`.

Each s-node is labeled with one or more functions of time, and probably some other process parameters. Each function denotes a property of a physical component of the system being modeled. These properties are affected by a change in system state. While the s-nodes represent sensors and actuators of a CPS being modeled, it

is these time-dependent functions that model the system dynamics and the name DSCG. In this case study, the functions representing product and component properties, were not used. However, these function are essential when a DSCG is used in creating a realistic simulation of the system.

For convenience, a DSCG is usually presented as a collection of sub-graphs, each corresponding to one or more components, or even the state of a component, of the CPS. Given the distributed and connected nature of a CPS control software, these subgraphs are connected. For example, one subgraph might indicate conditions for turning a pump ON. Another subgraph might use the ON state of the pump as a condition to change the state of another component.

Example 1 Consider a system *S*, as in Fig. 4a, consisting of the following components: a pump, a valve, and two water tanks. PLC 1 controls the pump while PLC 2 controls the valve. Level sensors at each tank communicate with the PLCs as shown. Each tank can be in any of four states: `LowLow (LL)`, `Low (L)`, `High (H)`, and `HighHigh (HH)`. The pump has two states: ON and OFF. The valve that connects pump to Tank B can be in one of two states: `O(pen)` and `C(losed)`.

Now suppose that the design of *S* requires the following conditions to govern the pump. The pump is started, i.e., its state changed from `OFF` to `ON`, when Tank A is not in state `LowLow`, Tank B is not in state `HighHigh`, and the valve is open (`O`). The pump state is changed from `ON` to `OFF` whenever Tank B is in the `HighHigh` state or Tank A is in `LowLow` state. Figure 4 shows a partial DSCG for *S* that captures the conditions that govern the pump operation. This partial DSCG has two *s-nodes* labeled `PON` and `POFF`, two *c-nodes* labeled Tank A and Tank B, and two *op-nodes* labeled $\wedge$ and $\vee$.

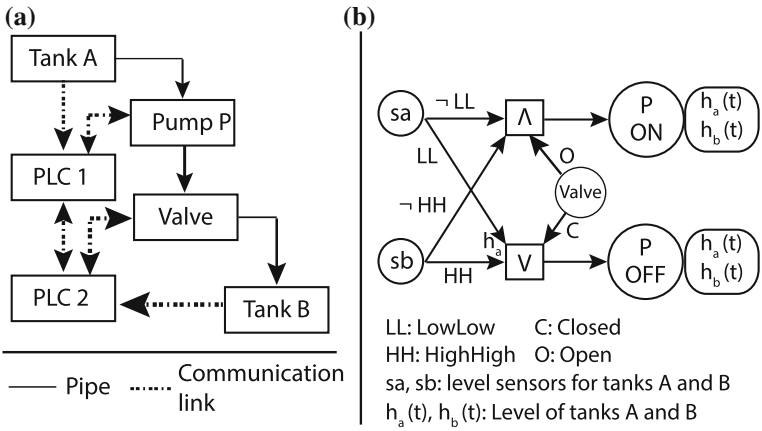

Fig. 4 **a** A subsystem of a CPS *S* consisting of two tanks, a pump, a valve, and two PLCs controlling the pump and the valve. **b** A portion of the DSCG for *S* depicting the conditions that govern pump operation and the time-dependent properties of the affected components. Conditions that govern the state change of the valve are not shown here

The state of pump P effects the water level in tanks A and B. Thus, when P is in state ON, the water level in tank A, $h_a(t)$, reduces while that in tank B, $h_b(t)$ increases. Functions $h_a(t)$ and $h_b(t)$ depend on the mechanical properties of the pump and dimensions of the two tanks. The discrete state assigned to a tank depends on the values of the corresponding height function $h(t)$. ■

4.3 Attacker Models

For a CPS, one possible attacker model is a pair (T, O), where T is an attack type to realise objective O. The attack type could be of any of the types proposed earlier such as in [6, 8, 9, 17]. More complex attack types are possible. The objective is specified as a statement. For example, "*Damage generator A in a power grid*," or "*Damage pump P302 in a water treatment network*." A *cyber* attack is a sequence of actions, a procedure, initiated by the attacker where each action is initiated via a cyber component, such as a wireless link or a SCADA computer. When the attack is via a physical component, such as an explicit damage to a pump or the physical removal of a circuit breaker, it is considered a *physical* attack. The actions in an attack are selected and sequenced so as to model the attack type T and realise the objective O. Whether or not the attempted action sequence will realise the attacker objective depends primarily on the defense mechanism used in the CPS.

Example 2 Consider the following attacker objective and attack type for the system in Fig. 4: *Cause Tank B to overflow* using a *deception attack* [18]. The attacker uses the following procedure to achieve the objective.

(1) **Enter and capture**: Identify the wireless communication links and capture the link from Tank B to PLC 2.
(2) **Wait and listen**: Listen to the data transferred across the links. Wait until Pump B is ON, the valve is C(losed), and Tank B is close to entering the HH state, say, when it is in state H.
(3) **Deceive**: Regardless of the data input from the Tank B level sensor, send to PLC 2 a value that corresponds to H.
(4) **Wait and listen**: Continue monitoring the Tank B level sensor until a few minutes after it outputs a value that corresponds to HH. An overflow will occur if the pump has not been shut sometime after Tank B moves to HH. The exact time when the overflow occurs depends on the excess capacity in Tank B beyond that needed in HH.
(5) **Exit**: Exit from the system when satisfied that the overflow has occurred. ■

4.4 Impact Analysis

Each possible action in the attack needs to be analyzed for its possible impact. This analysis aids in identifying possible weaknesses in the current defense mechanism, and hence in making it more robust. The complete DSCG graph, or a simulation based on it, is traversed to determine the potential impact of each action; the actual impact can be determined by implementing the attack on a realistic testbed as done in this case study. It is indeed possible that while the attacker objective may or may not be realized through the proposed actions, undesirable side effects might. It is best to perform this analysis from a pessimistic view. For example, instead of assuming that a given action, such as capture of a wireless link, is infeasible, it might be wise to assume that it is. Note that the impact of a cyber attack depends on the state of the system at the time of launch. However, this aspect is not considered in this paper.

Example 3 A brief sketch of impact analysis is presented next based on the actions described in Example 2. The focus here is on action "Deceive" as the impact of the other actions is relatively easy to determine.

At the time the deception action is initiated, the inputs to the OR ($\wedge$) node in Fig. 4b are: $\neg$ `LL`, `C(losed)`, and $\neg$ `HH`. In fact this is the correct state of the SWaT. However, depending on the rate of flow, Tank B will soon be in `HH` state while the input to PLC 2 will remain at H as determined by the attacker. At this time the deception action causes PLC 2 to incorrectly assume the state of the system; specifically the state of Tank B. This state divergence, i.e., the difference between the actual and the computed system states, remains until the attack is detected and the system reset.

Moving ahead, the incorrect state assumption by PLC 2 causes Pump B to remain `ON` as neither of the two conditions at the OR node is true. Over time, and unless there is an effective defense mechanism, Tank B will overflow.

The analysis now must continue with the remainder of the DSCG for S. Overflow of a tank might cause inconvenience or wasted water. It might also lead to more serious scenarios such a electrical short circuit and its impact on the control devices. ■

4.5 Design Update

Impact analysis will likely expose weaknesses in the CPS defense mechanism. In turn, CPS designers could then decide whether to remove the exposed weaknesses, or to let them remain perhaps because of the low probability of success of the attack that exposed the weakness, being successful, or due to reasons of economy.

Example 4 The analysis in Example 3 reveals a potential weakness in the defense against an attack on the Tank B level sensor. There are several defenses against this

attack. A hardware defense is to create a mechanical interlock that ensures (a) the pump is shut off automatically when Tank B is in state HH, and (b) an alarm is raised at the SCADA as well as physically near Tank B.

A software defense is also possible, though is more complex than the above hardware defense. PLC 2 could use a model that allows it to compute the water level in Tank B. The computed level could be compared with that received from the sensor. Any significant discrepancy is a cause for alarm. Several other possibilities exist, not discussed here, for enhancing the system defense against deception attacks. ■

5 Case Study

The design and analysis approach described above was used to analyze the defense mechanism of a Secure Water Treatment (SWaT) testbed. While the proposed procedure is intended to be applied during the CPS design process, in the case study reported here the procedure was applied on an operational system. As is often the case, SWaT was designed and built for correct operation. While the control algorithms in SWaT do account for component failures, they are not designed to detect and defend against cyber attacks. This aspect of SWaT makes it a useful subject to study the effectiveness of the DSCG-based modeling and analysis procedure.

5.1 Architecture of SWaT

SWaT is a testbed for water treatment. In a small footprint producing 5 gallons/h of filtered water, the testbed mimics a large modern water treatment plant found in cities. It is used to investigate response to cyber attacks and experiment with novel designs of physics-based and other defense mechanisms. As shown in Fig. 5, SWaT consists of six stages labeled P1 through P6. Each stage is controlled by its own set of dual PLCs, one serving as a primary and the other as a backup in case of any failure of the primary.

Communications: Each PLC obtains data from sensors associated with the corresponding stage, and controls pumps and valves in its domain. Turning the pumps ON, or opening a valve, causes water to flow either into or out of a tank. Level sensors in each tank inform the PLCs when to turn a pump ON or OFF. Several other sensors are available to check the physical and chemical properties of water flowing through the six stages. PLCs communicate with each other through a separate network. Communications among sensors, actuators, and PLCs can be via either wired or wireless links; manual switches switch between the wired and wireless modes.

Stages in SWaT: Stage P1 controls the inflow of water to be treated by opening or closing a valve (not shown) that connects the inlet pipe to the raw water tank.

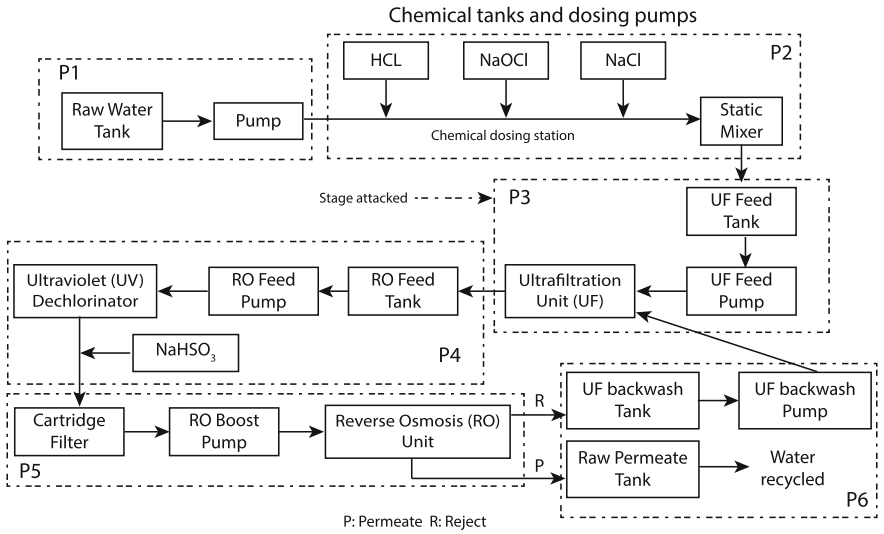

Fig. 5 Physical water treatment process in SWaT and attack point used in the case study. P1 though P6 indicate the six stages in the treatment process. The attack reported in this paper is on stage P3. Each stage is controlled by its own PLC connected to sensors and actuators. PLCs communicate among themselves and the SCADA computer via separate networks

Water from the raw water tank is pumped via a chemical dosing (stage P2) station to another UF Feed water tank in the stage P3.

In stage P3, a UF feed pump sends the water via UF membrane to RO feed water tank in stage P4. Here an RO feed pump sends the water through an ultraviolet dechlorination unit controlled by a PLC in stage P4. This step is necessary to remove any free chlorine from the water prior to passing it through the reverse osmosis unit in stage P5. Sodium bisulphate (NaHSO3) can be added in stage P4 to control the ORP.

In stage P5 the dechlorinated water is passed through a 2-stage RO filtration unit. The filtered water from the RO unit is stored in the permeate tank and the reject in the UF backwash tank. Stage P6 controls the cleaning of the membranes in the UF unit by turning on or off the UF backwash pump. The backwash cycle is initiated automatically once every 30 min and takes less than a minute to complete. Differential pressure sensors in stage P3 measure the pressure drop across the UF unit. A backwash cycle is also initiated If the pressure drop exceeds 0.4 bar indicating that the membranes need immediate cleaning. A differential pressure meter installed in stage P3 is used by PLC 3 to obtain the pressure drop.

Cyber attack points in SWaT: In this case study the wireless links between sensors and the corresponding PLCs are considered as the attack points. A pessimistic approach is taken implying that all wireless links are assumed to be vulnerable to cyber attacks. Initial experiments, not described here, revealed that

indeed, wireless communications in SWaT are vulnerable. In this study various level transmitters were considered as attack points.

5.2 Attacker Models

Attacker models are needed to understand the response of SWaT to malicious attacks. Table 1 lists four attack types and, for each type, one attacker objective. The attack types used here have been proposed by Cardenas et al. [9]. Note that a variety of attacks can be generated using each attacker model and attack type. Only one attack of each attack type is reported in this case study as an illustration of the design analysis procedure.

5.3 Modeling SWaT

This case study was performed on an operational system. While the complete design of SWaT is available to the authors, the DSCG model was created using actual ladder logic and structured text code [19] that resides in the six PLCs. The choice of PLC code, instead of using the design, was motivated by the desire to obtain an accurate model of SWaT.

SWAT model in terms of DSCG subgraphs consists of total of 12 sub-graphs. The sub-graphs are connected through links across the PLCs. Conditions governing the control of each pump and each motorized valve were modeled. Due to space limitations the DSCGs are not shown here.

5.4 Choice of Attacks and Impact Analysis

Detailed impact analysis was conducted using the 12 DSCGs developed from the PLC code. Each DSCG corresponds to conditions to change the state of an actuator in SWaT. Four cyber attacks were selected, one for each attack type in Table 1. Here only one implementation of the surge attack is described. The objective of the attack was to damage the UF unit. Assuming that an attacker knows the mechanical

Table 1 Attacker models used in the case study

Attack type	Attacker objective
Bias	Disturb chemical dosing
Covert	Affect dechlorination
Replay	Affect water quality
Surge	Damage the ultrafiltration unit

and chemical properties of the UF unit, and the overall structure of SWaT, attacks to damage the UF unit can be derived. These attacks would likely be successful when appropriate defense mechanisms do not exist.

The attacks in this case study were launched from the SCADA computer. These could also be launched in SWaT via the wireless network that connects the sensors to the PLCs. The actions needed to damage the UF unit are in Table 2 and numbered 1 through 4. The second from left column in the table lists attacker actions to achieve the desired objective. These actions are derived from the subgraph in DSCG that corresponds to pump P301. The rightmost column lists the consequence of each attacker action derived also from DSCG. Note that the attacker actions cause a discrepancy between the actual system state and the state known to PLC 3. The consequence of each attacker action, expressed in terms of conditions evaluated by PLC 3, are derived using this DSCG. The eventual impact on the UF unit is not derived using DSCG; it is derived through a series of arguments, not mentioned here, based on the mechanical properties of the UF membranes.

5.5 *Impact Analysis Summary*

A summary of all four types of attacks considered and their potential impact on SWaT appears in Table 3. Note that due to lack of hardware and software defense mechanisms, SWaT components such as UF and RO are likely to be damaged if these attacks were implemented and sustained for a long period. This claim of

Table 2 Impact analysis using a DSCG; attacker objective: damage the ultrafiltration unit

Attack	Actions	Consequence
1	Spoof messages going to PLC 3 by compromising the wireless link from the sensors	Attacker can send false data to PLC 3
2	Set the high pressure sensor PSH-301 to 2.0 Bar	System state: PSH301 > 2.5 Bar In PLC: PSH301 < 2.5 Bar Hence, in the absence of the attack, P301 should be turned OFF, but as the PLC has the incorrect state information, it does not turn P301 OFF
3	Set the differential pressure switch DPSH-301 to 0.3 Bar	System state: DPSH301 > 0.5 Bar In PLC: DPSH301 < 0.5 Bar Hence, in the absence of the attack, P301 should be turned OFF, but as the PLC has the incorrect state information, it does not turn P301 OFF
4	Set the differential pressure indicator DPIT-301 to 0.3 Bar	System state: DPIT301 > 0.4 Bar In PLC: DPIT301 < 0.4 Bar

Impact on SWaT: UF does not enter immediate backwash cycle; UF deterioration accelerated; UF is likely to be damaged if the attack persists for sufficient time. The time to damage the UF will depend on the incoming water quality and the properties of the membranes in the UF unit

Table 3 Summary of impact analysis on SWaT

Attack type	DSCG used	Outcome	Damage
Bias	p2_off	Dosing does not get activated to change the water properties	Water produced does not maintain desired chemical properties
Covert	p4_on	Water dechlorination does not take place for 10 min	Increased chances of damage to the RO unit
Replay	p5_on	Impure water gets into the RO unit permeate tank	No hardware damage
Surge	p3_off	Ultrafiltration unit damage accelerated due to delay in backwash	Increased chances of UF damage

p2, p3, p4, and p5 correspond to, respectively, three dosing pumps, pump P301, pump P401, and pump P501. Thus, p2_off, p3_off, p4_on, and p5_on refer to their respective DSCGs

damage is based on the mechanical properties of the membranes in the UF unit. Also, the claim of damage is being made cautiously as regular physical checks of water quality could enable attack detection prior to component damage.

5.6 Design Update

Based on the impact analysis described above, a detailed design of the defense mechanisms ought to be considered. Such a design has not been attempted so far. Nevertheless, a few potential hardware and software defense mechanisms are considered next.

Improving the security of the wireless connections is an obvious defense against all spoofing attacks. In the present context, more interesting defense mechanisms related to actions 2 through 4 in Table 2 are considered. These actions prevent urgent cleaning of the UF unit. However, the regular 30 min cleaning cycle will still be active. If the attack happens soon after a regular cleaning cycle, then the UF will be operating, with clogged membranes, for at most about 29 min. Thus, depending on the quality of the incoming water, and characteristics of the UF membranes, damage will likely occur.

One defense against the above attack is to install one or more water quality meters in the pipe that carries water from UF output to the RO feed tank. While several such a quality meters are available in SWaT but not immediately following the UF unit. Installation of additional water quality sensors will require the PLC code to be updated. Doing so will also update the corresponding DSCGs for any further impact analysis.

Another defense against the above, as well as other cyber attacks, is to have an *independent* network of sensors [20] that regularly check the health of the system, and especially of the critical system components such as the UF and RO units. These sensors do not communicate with the PLCs but have a *one way* communication with

the SCADA system. The sensors continuously check against violation of water quality constraints and raise appropriate alarms. The independence of the sensor network is crucial in this mechanism to be effective against cyber intrusions.

6 Novelty, Automation, and Scalability

Novelty: Given that a large number of approaches exist for modeling CPS, the novelty of the DSCG-based approach to impact analysis needs to be addressed. Approaches known to the authors and cited later in Sect. 2 do not explicitly include (a) the cyber and physical components of a CPS and (b) the conditions that affect the state of an actuator. Such an inclusion in a DSCG adds significant value in the early analysis of a CPS design. First, by creating a graph adjacency matrix from the system DSCG, one can easily and automatically, determine how many and which components of a system could be impacted when a given sensor is attacked. This information is highly valuable both for an attacker and the designer. It allows the attacker to design attacks for maximum system damage or disruption while offering valuable hints to the designer as to where to spend the effort in placing hardware/software defense mechanisms. Thus, for a designer this analysis offers economic arguments in favor or against adding specific security hardware/software. Though straightforward, this analysis is not possible using models such as Petri net [21] or graph based [1, 22]. Another straightforward outcome of a DSCG is the path condition for different attacks. Thus, by traversing a DSCG from a specific sensor node to any other node, one can determine what conditions are necessary for an attack to be successful or not successful. Note that either of the analyses mentioned above are valid for both cyber and physical attacks.

Automation: As in Fig. 2, a DSCG is created from a CPS design. This cannot be done without redesigning existing system design tools [23, 24]. However, given a version of a design schematic that labels various components, it is a matter of software design to map any CPS design to a DSCG. Note that a schematic will likely not include time functions that describe the property of each system component, e.g., electric generator, or a partial product, e.g., water flow rate out of a tank, on the state of one or more actuator states. Perhaps with the aid of a database of well known physical properties of components, the task of associating product and component state and product property functions could also be automated. While the difficulty of automating the entire DSCG construction will be best gauged when this task has been completed at least once, no significant technical hurdles seem to be on its way.

Scalability: The scalability of a DSCG-based design analysis approach depends on at least two factors: number of sensors and actuators in a CPS under design, and the selection of cyber and physical attacks. The number of nodes in a DSCG is linear in the number of sensors and actuators. Given a CPS design with N sensors and M actuators, the total maximum number of subgraphs in a DSCG is kM, where constant k depends on the number of discrete states of each actuator. Thus, the

number of subgraphs grows linearly with actuators. The number of actual nodes in each sub-graph depends on the number of components controlled by each actuator. Again, this is linear in the number of system components. The adjacency matrix is of size C^2, where $C = N + M$. Thus, for a system with, say, $C = 3000$, the adjacency matrix will have a total of 9 million entries. Fast algorithms for transitive closure [25] could be used to rapidly perform reachability analysis.

Given the attack model in this paper, the potential number of attacks is exorbitant when a component state is represented by a real number. However, in practice, continuous state space is often reduced drastically through discretization and engineering design so that only a few useful and critical states are considered during analysis. Doing so reduces the number of potential cyber attacks to grow linearly in the number of cyber components in a CPS and grows similarly in the number of physical components. Further reduction in the number of attacks to be used during the analysis phase is possible using arguments based on the economics of the system. A discussion on these aspects of attack space reduction is beyond the scope of this paper.

7 Summary, Discussion, Next Steps

A procedure to model a CPS at the design and operational stage is proposed. The procedure is based on Dynamic State Condition Graphs that capture the conditions used by control algorithms to change the state of individual CPS components. The proposed procedure has been applied to study the vulnerabilities in the software and hardware design of a modern and realistic water treatment system. The analysis revealed several weaknesses in the system design. While the system was designed to function correctly, security was a minor factor in the design. Thus, the DSCG-based procedure helped in identifying various weaknesses and hinting at software and hardware means for their removal.

The use of DSCGs presents a simple and practically usable procedure to assess the defenses of a CPS. Simplicity, and hence its ease of use, is a key characteristic of the procedure. The case study presented in this paper offers a glimpse into how the notion of "Security by Design" can be realised in practice. The approach is realistic and does not rely on any form of abstraction such as that found in linear control flow models of systems [6, 26, 27]. Further, the graphical nature allows partial automation to understand how an attack progresses through a CPS.

The analysis presented in the case study in this paper was done manually. The graphs, not shown in this paper, are constructed using a Python program but the impact analysis was done manually. The analysis procedure needs some automation for it to be applicable in the design of realistic systems. However, doing so requires a clear understanding of component semantics such as when does a component fail. DSCGs could become an even more powerful tool once they are enhanced with physical operational constraints of each device included in the model.

Acknowledgments Thanks to: Nils Tippenhauer, Daniel Daniele Antonioli for demonstrating the feasibility of attacking wireless links between sensors and PLCs in SWaT; Kaung Myat Aung for assisting in the validation of constraints in DSCG; and Ivan Lee, Mark Goh, and Angie Huang for their constant assistance without which this research would not be possible.

References

1. Jajodia, S., Noel, S.: Advanced cyber attack modeling, analysis, and visualization. Technical Report AFRL-RI-RS-TR-2010-078. Final Technical Report, George Mason University (2010)
2. Abadi, M., Budiu, M., Erlingsson, U., Ligatti, J.: Control-flow integrity principles, implementations, and applications. ACM Trans. Inf. Syst. Secur. **13**(1), 4:1–4:40 (2009)
3. Chen, B., Kalbarczyk, Z., Nicol, D.M., Sanders, W.H., Tan, R., Temple, W.G., Tippenhauer, N.O., Vu, A.H., Yau, D.K.: Go with the flow: toward workflow-oriented security assessment. In: Proceedings of the 2013 Workshop on New Security Paradigms Workshop, NSPW '13, pp. 65–76 (2013)
4. Bhave, A., Krogh, B., Garlan, D., Schmerl, B.: View consistency in architectures for cyber-physical systems. In: Proceedings of 2nd ACM/IEEE International Conference on Cyber-Physical Systems (2011)
5. Sommestad, T., Ekstedt, M., Johnson, P.: Cyber security risks assessment with Bayesian Defense graphs and architectural models. In: 42nd Hawaii International Conference on System Sciences, pp. 1–20 (2009)
6. Kwon, C., Liu, W., Hwang, I.: Security analysis for cyber-physical systems against stealthy deception attacks. In: American Control Conference (ACC), 2013, pp. 3344–3349 (2013)
7. Pasqualetti, F., Dorfler, F., Bullo, F.: Attack detection and identification in cyber-physical systems. IEEE Trans. Autom. Control **58**(11), 2715–2729 (2013)
8. Wasicek, A., Derler, P., Lee, E.: Aspect-oriented modeling of attacks in automotive cyber-physical systems. In: Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE, pp. 1–6 (2014)
9. Cárdenas, A.A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y., Sastry, S.: Attacks against process control systems: risk assessment, detection, and response. In: ACM Symposium on Information, Computer and Communication Security (2011)
10. Kara-Zaitri, C., Keller, A., Barody, I., Fleming, P.: An improved fmea methodology. In: Reliability and Maintainability Symposium, 1991. Proceedings, Annual, pp. 248–252 (1991)
11. Li, J., Xuan, C., Shao, B., Ji, H., Ren, C.: A new connected device-based failure mode and effects analysis model. In: 2014 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), pp. 1–5 (2014)
12. Young, W., Stamp, J., Dillinger, J.: Communication vulnerabilities and mitigations in wind power SCADA systems. In: American Wind Energy Association WINDPOWER Conference Austin, Texas, pp. 1–15 (2003)
13. http://www.mathworks.com/products/simulink/
14. http://www.ni.com/labview/
15. Stouffer, K., Scarfone, J.F.K.: Guide to Industrial Control Systems (ICS) Security (2011)
16. Galloway, B., Hancke, G.: Introduction to industrial control networks. IEEE Commun. Surv. Tutorials **15**(2), 860–880 (2013)
17. Amin, S., Càrdenas, A., Sastry, S.S.: Safe and secure networked control systems under denial-of-service attacks. In: Hybrid Systems: Computation and Control. Proceedings of 12th International Conference (HSCC), LNCS, vol. 5469, pp. 31–45. Springer (2009)
18. Amin, S., Litrico, X., Sastry, S., Bayen, A.: Cyber security of water SCADA systems; Part I: analysis and experimentation of stealthy deception attacks. IEEE Trans. Control Syst. Technol. **21**(5), 1963–1970 (2013)

19. Allen-Bradley: Logix5000 Controllers Structured Text, Programming Manual, Publication 1756-PM007D-EN-P, Rockwell Automation (2012)
20. Sabaliauskaite, G., Mathur, A.P.: Intelligent checkers to improve attack detection in cyber physical systems. In: Proceedings of the 2nd IEEE International Workshop on Cyber Security and Privacy (CSP 2013), International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC 2013) Beijing, PRC (in press) (2013)
21. Chen, T., Sanchez-Aarnoutse, J., Buford, J.: Petri net modeling of cyber-physical attacks on smart grid. IEEE Trans. Smart Grid **2**(4), 741–749 (2011)
22. Bondavalli, A., Lollini, P., Montecchi, L.: Graphical formalisms for modeling critical infrastructures. In: Critical Infrastructure Security: Assessment, Prevention, Detection, Response. WIT Press Royal (2011)
23. Power plant design
24. Siemens: Programming with STEP 7 Manual, 05/2010, a5e02789666-01
25. Lidl, R., Pilz, G.: Applied Abstract Algebra, 2nd edn. Springer (1998)
26. Mo, Y., Hespanha, J., Sinopoli, B.: Robust detection in the presence of integrity attacks. In: American Control Conference (ACC), 2012, pp. 3541–3546 (2012)
27. Pasqualetti, F., Dorfler, F., Bullo, F.: Attack detection and identification in Cyber-Physical Systems, models and fundamental limitations. IEEE Trans. Autom. Control **58**(11), 2715–2729 (2013)