Using Process Invariants to Detect Cyber Attacks on a Water Treatment System

Sridhar Adepu, Aditya Mathur

Using Process Invariants to Detect Cyber Attacks on a Water Treatment System

Sridhar Adepu and Aditya Mathur

Singapore University of Technology and Design, Singapore, 487372.
sridhar_adepu@sutd.edu.sg, aditya_mathur@sutd.edu.sg

Abstract. An experimental investigation was undertaken to assess the effectiveness of process invariants in detecting cyber-attacks on an Industrial Control System (ICS). An invariant was derived from one selected sub-process and coded into the corresponding controller. Experiments were performed each with an attack selected from a set of three stealthy attack types and launched in different states of the system to cause tank overflow and degrade system productivity. The impact of power failure, possibly due to an attack on the power source, was also studied. The effectiveness of the detection method was investigated against several design parameters. Despite the apparent simplicity of the experiment, results point to challenges in implementing invariant-based attack detection in an operational Industrial Control System.

Keywords: Attack detection, Cyber attacks, Cyber Physical Systems, Industrial Control Systems, Secure water treatment testbed.

1 Introduction

An experimental investigation, referred to as EXP, was undertaken with the long term goal of designing robust defense mechanisms for an Industrial Control System (ICS) and to improve its resiliency. A short term goal in EXP is to understand how to detect cyber attacks against an ICS using state invariants across data obtained by a Programmable Logic Controller (PLC) from two or more sensors. Such an understanding leads to the inclusion of effective detection mechanisms inside process controllers and the addition of control actions when an attack is detected thereby improving system resiliency. The experiments were performed to understand the effectiveness of the proposed detection method in an operational mini-water treatment testbed, referred to as SWaT (Secure Water Treatment), that produces 5 gallons/minute of treated water.

Invariant: An "invariant" is a condition among "physical" and/or "chemical" properties of the process that must hold whenever an ICS is in a given state. Together, at a given time instant, measurements of a suitable set of such properties constitute the observable state of SWaT. In SWaT, these properties are measured using sensors and captured by the PLCs at programmable time instants, (set to

0.1-second in EXP). A few key advantages of invariant-based attack detection, implemented in a PLC, are as follows.

1. *Implementation context*: Detection method is implemented as a procedure and integrated directly into the PLC.
2. *Physical constraints*: The invariants are local to a PLC and based on the physics or the chemistry of the sub-process being controlled by the PLC.
3. *Detection method*: Detection is designed without reference to attacks and hence is *attack agnostic*. It is based on state related conditions that must hold either always during system operation, or when one or more components of the system is in a given state. For example, the level of water in a tank must be always between its lowest and highest points; this invariant must always hold during normal system operation. However, a pump must be in the ON state only when the source tank has water and the destination tank is not full; this is a state dependent invariant.
4. *Network traffic*: There is no additional load on the ICS communications network.

Research focus: RF1: Methods for detecting single-point cyber attacks based on invariants derived from physical properties of the process. RF2: Effectiveness of the invariant-based detection methods. RF3: Complexity and scalability of the detection methods and their impact on the operation of a PLC.

Related work: In a survey [8] the detection techniques are classified as follows: misuse/signature-based intrusion detection, anomaly-based intrusion detection, and statefull protocol analysis. The process invariants based approach proposed in this paper does not fall in any of these three categories. Further, as implemented in the case study described here, the proposed approach differs from those mentioned above in several ways including the fact that it does not use network-based anomaly detection. Pros and cons of the proposed approach against others are discussed in Section 4. A common aspect of the techniques described in [8] is that they are employed in the network used for communications among the PLCs and SCADA. The primary goal of these techniques is to detect any intrusion into the ICS by analyzing network traffic from different points of view. For example, in [3] security specifications are defined for smart meters and a security policy for the Advanced Metering Infrastructure.

The use of invariants in CPS is not new. In [12] the authors used invariants as a unified knowledge model for CPS. Some authors refer to invariants as “attack symptoms” and have used these to detect attacks in intrusion detection [9]. Stability of smart-grid has been studied using invariants [5]. The key contribution in this paper is the derivation and use of invariants using the physics of a water treatment plant and its application in detecting novel attacks.

A specification-agnostic technique for the detection of cyber attacks in PLCs is described in [7]. It is noted that this technique does not fall in any of the three categories mentioned above. Data is collected from the PLCs via network monitoring, and an autoregressive method is used to model specific process variables.

Such a model is then used to detect whether the value of a process variable is suspicious. Another technique uses the physics of an ICS, using conditions similar to invariants, to detect cyber attacks [10]. Here the authors analyze network traffic focusing on "harmful" command streams. Physical constraints are integrated into an intrusion detection framework. An example of a boiler is described where out of bound values are checked against predefined constraints.

Significant work exists in detecting anomalies in network traffic in ICS across PLCs, sensors, actuators and SCADA subsystems. One such technique is based on CUSUM used for change point detection. This non-parametric method has been applied to detect network intrusions [14]. While these techniques are found effective in environments in which they were assessed, in EXP it was decided to instead use only the process property based invariants to detect anomalies arising due to a cyber attack. Doing so avoids making assumptions on probability distributions of process data. Indeed, making use of invariants is perhaps appropriate when a real or simulated process is available for experimentation, and not necessarily when only data from such process is available as for example in [7, 15].

Intermittent cyber attacks: Studies on the impact and detection of intermittent, or pulse, attacks on networks have been reported. In [16] the authors considered the impact of low-rate denial of service (LDoS) attacks on networks. A wavelet based method was proposed for detecting such attacks. In [13] the authors focused on pulsing DoS attacks and their impact on peak bandwidth. The experiments described in this paper are aimed at investigating how intermittent cyber attacks on an ICS, and not on networks, can lead to undesirable behavior and the difficulty of detecting them using invariants. The authors of the current study are not aware of any experiment that investigates the impact of intermittent attacks on ICS.

Contributions: (a) Invariant-based approach for attack detection in a specific ICS. (b) Dependence of the invariant-based approach on the system state, and several other parameters, when an attack is launched. (c) Impact of attack detection study on the design of the software and hardware of a specific ICS.

Organization: Section 2 describes the method used and the experiments conducted. A brief introduction to SWaT is in this section. Results from the experiment appear in Section 3. Discussion on various aspects of attack detection in an ICS and design challenges, are in Section 4. Conclusions and plans for further experimentation are in Section 5.

2 Method

2.1 Context: The SWaT testbed

SWaT is a fully operational scaled down water treatment plant for research in the design of ICS resilient to cyber and physical attacks. In a small footprint producing 5 gallons/minute of doubly filtered water, this testbed mimics large

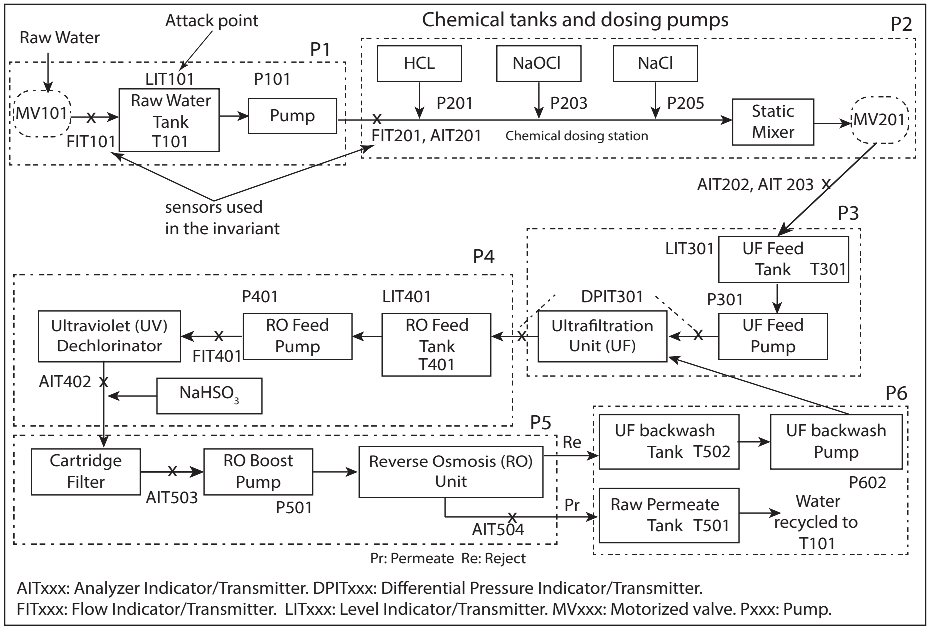

Fig. 1. Water treatment in SWaT: P1 through P6 indicate the six stages in the treatment process. Arrows denote the flow of water and of chemicals at the dosing station.

modern plants for water treatment such as those found in cities. The testbed is available for investigating the response to cyber-attacks and for conducting experiments with novel designs of physics-based and other attack detection and defense mechanisms.

Water treatment process: The treatment process (Figure 1) in SWaT consists of six distinct and cooperating sub-processes P1 through P6. Each sub-process, referred to as a *stage*, is controlled by an independent PLC. Thus, six PLCs work in concert to control the entire process. Control actions are based on the system state estimated by the PLCs using data from sensors.

Stage P1 controls the inflow of water to be treated, by opening or closing a motorized valve that connects the inlet pipe to the raw water tank. Water from the raw water tank is pumped via a chemical dosing station (stage P2) to another UF (Ultra Filtration) feed water tank in stage P3. A UF feed pump in P3 sends water via the UF unit to RO (Reverse Osmosis) feed water tank in stage P4. Here an RO feed pump sends water through an ultraviolet dechlorination unit controlled by a PLC in stage P4. 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.

Communications: Each PLC obtains data from sensors associated with the corresponding stage, and controls pumps and valves in its domain. Ultrasonic level sensors in each tank inform the PLCs of water level in the corresponding tank. 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 allow switch between the wired and wireless modes.

Attacking SWaT: The wireless network in SWaT connects PLCs to sensors, actuators, and to the SCADA server and an engineering workstation. Attacks that exploit vulnerabilities in the protocol used, and in the PLC firmware, are feasible and could compromise the communications links between sensors and PLCs, PLCs and actuators, among the PLCs, and between PLC and SCDA and the historian. Having compromised one or more links, an attacker could use one of several strategies to send fake state data to one or more PLCs, or simply do reconnaissance for a possibly subsequent attack.

2.2 Experiments

System state *Components and states*: Each PLC in SWaT controls one or more actuators such as a pump. A *Local Component Set* consists of components whose state is directly sensed by a PLC and the actuators that the PLC controls. The actions of a PLC depend on the state of the components in its LCS, and might also depend on the state of components in the LCS of one or more other PLCs. In the latter case a PLC can communicate with the other PLCs via the communication network to retrieve the required state data. The union of LCS for each PLC constitutes the *Global Component Set (GCS)*. The *local* state of SWaT is comprised of the respective observable states of components under direct control of a single PLC, i.e., that of its LCS. A collection of local observable states of all six PLCs in SWaT constitutes its *global* state. It is important to note that only the observable properties of the process and its components are included, and used for attack detection, in the local and global states.

State set: As shown in Table 1, three distinct local observable states were selected for the experiments reported here. These are the states when an attack is launched. In $\mathbf{S}_0^1$ the level of Tank T101 is constant during the attack. In $\mathbf{S}_0^2$ the level of T101 is increasing which happens when the level goes below a pre-determined value. In $\mathbf{S}_0^3$ the level in Tank T101 is decreasing which could happen because tank T201 in the sub-process in P3, is below a pre-determined level. These three states of T101 are marked as A, B, and C in Figure 2.

Attack design *Attack types*: In this work the focus is on attack detection assuming that an attacker succeeds in launching it. While paths through a system used by an attacker to enter by exploiting a system vulnerability, e.g., design flaw in a PLC [6], is an important topic, it is not the focus of this work. The

Table 1. State of P1 at the time of attack launch.

System State	Sensor[†]	State	Description
$\mathbf{S}_0^1$	**LIT101**←	HH[‡]	T T101 is full
	MV101	Closed	No flow into T101
	P101	OFF	No outflow from T101
$\mathbf{S}_0^2$	**LIT101**←	L	T101 is not full
	MV101	open	Water flows into T101
	P101	OFF	No outflow from T101;
$\mathbf{S}_0^3$	**LIT101**←	H	T101 is full
	MV101	Closed	No flow into T101
	P101	ON	Flow out of T101

[†] Sensors not listed are not used during attack detection. As marked, LIT101 is attacked. [‡]Tank states: HH=1000mm, H=800mm, L=500mm, LL=250mm.

number of possible attacks on an ICS is exorbitantly large. Three distinct types of attacks were selected, namely, fixed bias (FB), fixed bias intermittent (FBI), and variable bias intermittent (VBI). The choice of these three attack types was motivated by (a) attacks used in [4] to study the effectiveness of a statistical technique in detecting cyber attacks in a chemical process, and (b) a series of thought experiments aimed at designing attacks that might be difficult to detect in certain specific states of SWaT.

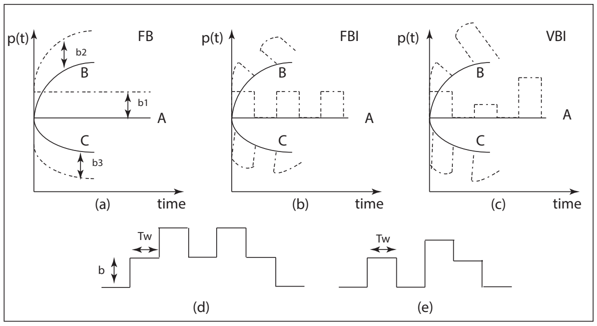

Fig. 2. Attack types (a) Fixed Bias (FB), (b) Fixed Bias Intermittent (FBI), and(c) Variable Bias Intermittent (VBI), superimposed on physical property p(t) as indicated by the dotted lines. In FB and FBI, the attacks follow the change in property p(t) in an attempt to avoid detection. (d) FBI; bias and pulse width are fixed. (e) VBI; bias b and pulse width T_w are varied.

Stealthy attacks: A stealthy attack on a CPS is one that remains undetected unless special detection mechanisms are in place. Such attacks have been studied in the context of CPS [4]. While non-stealthy attacks are possible in SWaT, all attacks launched in the experiment reported here were stealthy as the system software and hardware was unable to detect these until a system damage had occurred [1].

Attack: As shown in Table 2, several attacks types were examined within each state. For example, in NPF (No Power failure), an attack is launched after the PLC has initialized itself with the current state of its local components. In PF, an attack is launched just prior to the PLC starting to initialize itself and hence does not *yet* have information about its local components. In addition, the bias b used in an attack as well as the attack pulse width (T_w) were varied. The duration of attack detection (n), i.e., the number of sensor readings used to compute the invariant, was fixed at 10, where one reading is obtained every second. A selection of one value from each set in Table 2 served to define an attack. For example, an FBI attack was launched when SWaT was in state $\mathbf{S}_0^1$, attack bias was larger than 4, pulse width was 8, and the system was operating in a stable state, i.e., there was no power outage in the immediate past. Thus, a total of $3 \times 2 \times 3 \times 2 \times 2 \times 1 = 72$ experiment combinations exist. Additionally, a large number of trial runs had to be conducted to obtain reasonable values of bias, width of the attack pulse, and parameter ϵ mentioned later in Eq. 4.

Table 2. Summary of states and parameters used in the experiments.

State or parameter	**Values**	**Comments**
Actuator and tank states	$\{\mathbf{S}_0^1, \mathbf{S}_0^2, \mathbf{S}_0^3\}$	States of various components of SWaT set prior to launching an attack; details are in Table 1.
Power outage	{NPF, PF}	Attack launched during normal operation (NPF) and immediately before or after power outage (PF)
Attack type	{FB, FBI, VBI}	Fixed bias (FB), fixed bias intermittent (FBI), and variable bias intermitent (VBI)
Bias (b)	$\{\{> 4\text{mm}\}, \{< 4\text{mm}\}\}$	Bias used each of the three attack types.
Pulse width (T_w)	{8, 13}	Width of the attack pulse (in seconds)(Figure 2 used in FBI and VBI.
Detection duration (n)	10	Number of sensor readings used prior to announcing a decision on whether the sensor is under attack or not.

Attack procedure: The following general procedure was used to launch cyber attacks in EXP.

1. Identify the *tag* to be manipulated in the attack; a tag is a memory location where a PLC saves the received sensor data.
2. Compromise the wireless link between the SCADA computer and PLCs.

3. Manipulate the *tag* by setting its value different from that received by the PLC. In the absence of any hardware or attack detection logic in the PLC code, the PLC assumes the manipulated value to represent the true state of the component that corresponds to the sensor whose output is manipulated.

2.3 State estimation

Let x denote property p and y its measurement. $y(k)$ denotes the sensor measurement for $x(k)$ at instant k. $\hat{x}(k)$ is an estimate of $x(k)$. In the absence of sensor errors and no cyber attacks, $\hat{x}(k) = x(k) = y(k)$. In EXP the water level in tank T101 was considered as p. The level in T101 is measured by sensor LIT101 (Figure 1) that was assumed to be under attack. Sensors FIT101 and FIT201 measure, respectively, water flow into and out of T101. These flow rates are denoted as $u_i(k)$ for inflow, and $u_o(k)$ for outflow. In SWaT, each PLC obtains sensor data at 0.1 second intervals though for detecting an attack data was sampled from LIT101 every second as smaller sampling intervals did not offer any benefit in attack detection in trial runs.

In EXP the attacker intent was to cause tank T101 to overflow and degrade the performance of SWaT so that it produces less water than its normal capacity. This intent was to be realized by attacking the level sensor LIT101 that measures and reports $x(k)$ to PLC 1 in stage P1. $\hat{x}(k+1)$ was computed using methods M1 and M2 described next.

Method M1: Open loop: In this method $\hat{x}(k+1)$ is estimated using $\hat{x}(k)$ as follows.

$$\begin{aligned}\hat{x}(0) =& y(0)\\ \hat{x}(k+1) =& \hat{x}(k) + \alpha(u_{in}(k) - u_{out}(k))\end{aligned} \tag{1}$$

where α converts flow rate to the change in the level of T101 using the physical dimensions of the tank.

Method M2: Closed loop: In this method $\hat{x}(k+1)$ is estimated using $y(k)$ as follows.

$$\begin{aligned}\hat{x}(0) =& y(0)\\ \hat{x}(k+1) =& y(k) + \alpha(u_{in}(k) - u_{out}(k))\end{aligned} \tag{2}$$

Note that in M1 the estimate of tank level is updated using previous state estimate with the initial value obtained from the sensor L101. In M2 the tank level is estimated using sensor values. In trial experiments it was observed that the open loop method is not suitable for deriving an invariant as the method does not account for the change in system dynamics. Hence M1 was abandoned.

2.4 Invariants

At time instant $k+1$, the water level in T101 depends on the level at time k and the inflow and outflow at instant k. This relationship is captured in the following idealized model of the tank,

$$x(k+1) - x(k) = \alpha(u_i(k) - u_o(k)), \tag{3}$$

where (3) assumes perfect sensors which is not true in practice. Hence, to derive a practically usable invariant, SWaT was run several times without any attacks to estimate the mean μ_d and the standard deviation σ_d of $d = (\hat{x}(k) - y(k))$ over several runs, i.e., the mean and variance of the difference between the estimated tank level $\hat{x}(k)$ and its measured value $(y(k))$. In these runs $\hat{x}(k)$ was computed using the closed loop method M2 as in Eqn. 2. Based on Eqn. 3, the statistics obtained experimentally, and converting the true states to their estimates, the following conditions were derived to test whether or not sensor LIT101 is under attack.

$$\begin{aligned}\frac{\sum_{i=1}^{n}(\hat{x}(i) - y(i))}{n} &> \quad \epsilon, \text{ under attack}, & (4)\\ &\leq \quad \epsilon, \text{ normal}. & (5)\end{aligned}$$

In the conditions above, the average of the difference between the estimated and the measured tank levels is tested against ϵ. Thus, a decision whether or not LIT101 is under attack is taken from n sensor readings. Selection of n ought to be done carefully as it impacts the detection effectiveness. In EXP n, was set to 10. As described earlier, based on trial runs of SWaT without attacks, ϵ was set to 0.55. Code that implements attack detection using the invariant in Eqns.4 and 5 was added to the control algorithm already built into PLC 1.

3 Results

Data obtained from the experiments, and its analyses, are reported in the following.

3.1 Detection effectiveness and impact

In each of the three states when the attacks were launched (Table 1), there were six attack modes, namely, FB-PF, FB-NPF, FBI-PF, FBI-NPF, VBI-PF, and VBI-NPF. The attack detection results are summarized as follows.

1. Detection of attacks launched in FB-PF mode, and the system response, was found to be independent of T_w. However, it does depend on bias b and the initial state when the attack was launched. This happens because the PLC looses prior state data $(y(k))$ from LIT101. Hence, it initializes $\hat{x}(0)$ by obtaining $y(0)$ from LIT101 which is under attack. Thus, the initial state estimate is the tank level indicated by the attacker, i.e., (actual tank level+b), and not the actual tank level for T101. This happens regardless of

the initial states of SWaT, i.e., $\mathbf{S}_0^1$, $\mathbf{S}_0^2$, and $\mathbf{S}_0^3$. From this point onwards, the PLC computes the remaining values of $\hat{x}(k+1)$ using the incorrect $\hat{x}(0)$. As explained next, the response of SWaT now depends on the initial state and $\hat{x}(0)$. As there exist several variations of the attack depending on the value of b, four sample cases are discussed next.

Case 1: Initial state=$\mathbf{S}_0^1$, $\hat{x}(0)$ <L. [*Attack not detected; tank overflow*] PLC opens MV101 and water starts flowing into T101. The invariant in Eq.4 is computed over the following 10 seconds. However, the condition for attack detection is false and hence the attack is not detected. After some time T101 overflows.

Case 2: Initial state=$\mathbf{S}_0^1$, $\hat{x}(0)$ >L. [*Attack not detected; performance degradation*] In this case the PLC does not open MV101. Assuming that the ultrafiltration process is active, the level in T301 is reducing. When this level falls below H then pump P101 will be started by PLC 1. This causes tank level of T101 to decrease gradually while LIT101 continues to report the injected value. Eventually T101 becomes empty, P101 is stopped and water stops flowing into T301. This will cause the level in T301 to drop and eventually stop P301 and the ultrafiltration process. This event will eventually lead to the RO process stopping as there is no water to be filtered. Thus, this attack leads to a reduction in system performance.

Case 3: Initial state=$\mathbf{S}_0^2$, $\hat{x}(0)$ <L. [*Attack not detected; overflow*] Given that the bias is fixed and the attacker is following the water level trajectory in the tank, MV101 will continue to be open until LIT101 indicates HH. However, the attacker can control b such that the level indicated by attacked LIT101 is much lower than the tank level at attack launch. Thus, if tank level is $L_1 < L$ immediately prior to the attack, and $b = -200$, then T101 will overflow because the buffer in T101, beyond level HH, is only 100mm. In this attack scenario, if $b > 0$ then the attack will not be detected and there will be no overflow as the PLC will shut MV101 when the injected value of LIT101 reaches HH.

Case 4: Initial state=$\mathbf{S}_0^3$, $\hat{x}(0)$ >H. [*Attack not detected; no damage*] In this case when the injected LIT101 value reaches L, MV101 will open and the tank level will start to rise. Thus, the attack is not detected while the level is decreasing. When the level begins to rise, the attacker will need to change the bias to remain undetected as the scenario now is similar to the one in case 3. For other values of b the attack is not detected and no damage done.

2. In the remaining five modes with b> 4, all 30 attacks were detected. Several experiments were performed to investigate the impact of b and T$_w$. With b= 3 and T$_w$= 8, 40% of the attacks launched were detected and the remaining not detected. With b= 3 and T$_w$= 13, 10% of the attacks were detected, and the remaining not detected. With b< 3 none of the attacks were detected.

3.2 Selection of n and ϵ

The experiments revealed the importance of selecting appropriate values of n and ϵ used in the invariant to decide whether or not LIT101 is under attack. Data from the experiments was used to investigate the relationship between n, ϵ, and false alarms. Such investigation is also needed, and is underway, to understand the relationship between n, ϵ and the attack detection effectiveness.

Figure 3 shows how false alarms depend on n and ϵ. The data in the plots in Figure 3 was collected over a period of 40 minutes of SWaT operation in NPF mode. It is clear that the false alarms decrease as n and ϵ increase. For $\epsilon = 0.5$, the best value of n lies around 6. Similarly, for $n = 3$ the best value of ϵ is around 0.65. Thus, $n = 6$ and $\epsilon = 0.65$ appears to be the best combination if minimization of false alarm rate is the objective. However, attack detection rate also reduces with increasing n as well as with increasing ϵ. Thus, additional experiments need to be conducted to find optimal values of n and ϵ that maximize the attack detection rate while minimizing the false alarm rate.

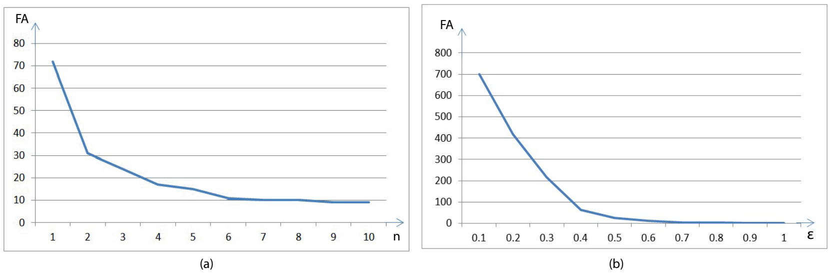

Fig. 3. False alarms (a) vs n for $\epsilon = 0.5$, (b) vs ϵ for $n = 3$.

4 Discussion

RF1: Two methods, namely, M1 and M2 are proposed for computing invariants to detect cyber attacks in ICS. Both methods can be used to derive invariants from the properties of the process that could be attacked. Experiments were conducted using only M2 as M1 was not found suitable for use in a system with changing dynamics.

RF2: In all experiments, attacks were launched when SWaT was in one of three system states, namely, $\mathbf{S}_0^1$, $\mathbf{S}_0^2$, and $\mathbf{S}_0^3$. Within each set there were two sub-states: without system reset (NPF) and soon after system reset (PF) such as what might happen after power is removed from the system. In NPF, the attack was launched soon after SWaT was started but all PLC's had executed their respective control algorithms at least once. In PF, the attack was launched and the PLC reset prior to it completing n cycles of code execution, where n is the attack detection window. Results indicate that attack detection becomes challenging when the attack is launched during PLC reset. Resetting a PLC causes it to loose the prior state information. Thus, if an attack detection algorithm must initialize its knowledge of the system state soon after the reset operation then doing such initialization from the sensor ought to be avoided as the attacker might have already compromised the sensor. Another approach could be to obtain the initial state of the process from the historian where all sensor data is saved. Certainly, this approach is advisable assuming that the sensor value saved in the historian is the true state value and not the one sent by an attacker. Thus, even in the case of a perfect attack detection algorithm, sensor data sent to the historian could be from an attacker if the attack was not detected prior to the reset operation.

RF3: Derivation and implementation of the invariants in Eqns. 4-5 was relatively straightforward. The invariant was implemented in Structured Text, a commonly used programming language for PLCs. Adding this code to the existing code in PLC 1 had negligible impact (at most 3milli-seconds) on the time to execute one scan cycle. It is not clear what will be the impact of adding code that implements traditional schemes, e.g. Kalman filter or Lurenberger observer, for removing noise from sensor data before it is used in computing the invariant. However, one could complement a PLC with additional hardware, say based on FPGA, that encodes the invariant and is connected to the main SCADA workstation to send alerts. Doing so would not add computational load to the PLC.

Generality of invariant based detection: Invariants derived in EXP capture the dynamics of tank level. In SWaT, several other invariants exist, but were not derived. These include, invariants that relate water properties such as pH, ORP, and conductivity as water flows across the chemical dosing station, UF, and RO units. Such invariants are important to detect cyber attacks aimed at affecting properties of water within a plant as well that coming out of a plant. However, such invariants also depend on the chemical and physical properties of the units involved. For example, the relation between the pH of water entering the RO unit and that coming out as permeate, depends on the physical properties of the membranes in RO. The time dependent nature of such properties requires tuning of any parameters used in the invariant to reduce false positives. It is evident that while for a specific sub-process in any ICS, one can define a process invariant, many such invariants exist in one plant.

Multiple point attacks: The cyber attacks considered in EXP are single point. The detection mechanism proposed in this work can be thwarted when the attacker has access to the sensors used in the invariants. For example, in Eqns. 1-2, sensors FIT101 and FIT201 are used to measure the inflow and outflow rates. The detection method can fail when these two flow sensors are compromised. Thus, to detect attacks on multiple sensors, one needs additional state information that will likely be derived from one or more sub-processes in an ICS that are not under attack. Doing so may or may not detect attacks, and even when detected, the detection might be delayed based on the state of the sub-processes.

Emerging design challenges: The following key design parameters were identified in EXP: n and ϵ in Eqn. 4; A_w: number of contiguous sensor inputs used by the control logic in the PLC to decide whether to initiate an action on an actuator; T_w: pulse width; and b: bias. n, A_w, and ϵ can be controlled in the software that implements the attack detection and control algorithms in a PLC. However, T_w and b are controlled by the attacker. Given the knowledge of n, ϵ, and A_w, the attacker can adjust T_w and b and succeed in causing SWaT malfunction, and especially so if the attack is launched before a PLC has been able to reset itself, i.e., in the PF mode. Parameter R_w can be controlled to some extent by selecting appropriate actuators. Thus, a key research question arises: How should one determine the most appropriate values of A_w, n, and ϵ given the uncertainty in T_w and b? This question assumes important in light of the fact that sensors could lead to spurious data and that the error profile of a sensor may change over time requiring the D_w and A_w to be retuned.

5 Conclusions and future work

An experimental investigation was undertaken to understand the effectiveness of attack detection using process invariants. The experiments were performed on an operational water treatment system. One water level sensor was selected as the target of the attacker. An invariant was derived from the dynamics of water flow into and out of a tank. Results from the experiments clearly indicate the strengths and limitations of the invariant-based approach for attack detection. Several ICS design parameters were identified. The appropriate values of these parameters can be selected by the designer and depend on system dynamics. Selection of parameter values is a subject of study by itself and needs to be taken seriously to avoid false positives. Note that it is possible for an attacker to bypass the detection method if the parameter values are known.

While the experiments indicate that the invariant-based detected method is effective in detecting a variety of attacks, no claims are made regarding the detection effectiveness in other domains such as power and transportation. A theoretical study is needed to better understand why physics-based invariants are, or are not, able to detect the attacks.

Acknowledgements

Kaung Myat Aung for assistance in conducting the experiments. This work was supported by research grant 9013102373 from the Ministry of Defense and NRF2014-NCR-NCR001-040 from the National Research Foundation, Singapore.

References

1. S. Adepu and A. Mathur. An investigation into the response of a water treatment system to cyber attacks. In *Proceedings of the 17th IEEE High Assurance Systems Engineering Symposium, Orlando*, January 2016.

2. J. Beaver, R. Borges-Hink, and M. Buckner. An evaluation of machine learning methods to detect malicious SCADA communications. In *12th International Conference on Machine Learning and Applications (ICMLA)*, volume 2, pages 54–59, Dec 2013.
3. R. Berthier and Sanders. Specification-based intrusion detection for advanced metering infrastructures. In *17th IEEE Pacific Rim International Symposium on Dependable Computing*, pages 184–193, Oct 2011.
4. A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: Risk assessment, detection, and response. In *ACM Symp. Inf. Comput. Commun. Security*, 2011.
5. A. Choudhari, H. Ramaprasad, T. Paul, J. Kimball, M. Zawodniok, B. McMillin, and S. Chellappan. Stability of a cyber-physical smart grid system using cooperating invariants. In *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual*, pages 760–769, July 2013.
6. ICS-CERT Advisories `https://ics-cert.us-cert.gov/advisories`.
7. D. Hadžiosmanović, R. Sommer, E. Zambon, and P. H. Hartel. Through the eye of the PLC: Semantic security monitoring for industrial processes. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 126–135, New York, NY, USA, 2014. ACM.
8. S. Han, M. Xie, H.-H. Chen, and Y. Ling. Intrusion detection in cyber-physical systems: Techniques and challenges. *IEEE Systems Journal*, 8(4):1049–1059, Dec 2014.
9. S.-W. Hsiao, Y. Sun, M. C. Chen, and H. Zhang. Cross-level behavioral analysis for robust early intrusion detection. In *IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 95–100, May 2010.
10. C. McParland, S. Peisert, and A. Scaglione. Monitoring security of networked control systems: It's the Physics. *IEEE Security Privacy*, 12(6):32–39, Nov 2014.
11. R. H. Niazi, J. A. Shamsi, T. Waseem, and M. M. Khan. Signature-based detection of privilege-escalation attacks on Android. In *2015 Conference on Information Assurance and Cyber Security (CIACS)*, pages 44–49, Dec 2015.
12. T. Paul, J. Kimball, M. Zawodniok, T. Roth, and B. McMillin. Invariants as a unified knowledge model for cyber-physical systems. In *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, pages 1–8, Dec 2011.
13. R. Rasti, M. Murthy, N. Weaver, and V. Paxson. Temporal lensing and its application in pulsing denial-of-service attacks. In *IEEE Symposium on Security and Privacy (SP)*, pages 187–198, May 2015.
14. A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *Signal Processing, IEEE Transactions on*, 54(9):3372–3382, Sept 2006.
15. G. Thatte, U. Mitra, and J. Heidemann. Parametric methods for anomaly detection in aggregate traffic. *Networking, IEEE/ACM Transactions on*, 19(2):512–525, April 2011.
16. Z.-j. Wu, L. Zhang, and M. Yue. Low-rate DoS attacks detection based on network multifractal. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–10, 2015.