Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems

Robert Mitchell and Ing-Ray Chen, *Member, IEEE*

***Abstract*—In this paper we analyze the effect of intrusion detection and response on the reliability of a cyber physical system (CPS) comprising sensors, actuators, control units, and physical objects for controlling and protecting a physical infrastructure. We develop a probability model based on stochastic Petri nets to describe the behavior of the CPS in the presence of both malicious nodes exhibiting a range of attacker behaviors, and an intrusion detection and response system (IDRS) for detecting and responding to malicious events at runtime. Our results indicate that adjusting detection and response strength in response to attacker strength and behavior detected can significantly improve the reliability of the CPS. We report numerical data for a CPS subject to persistent, random and insidious attacks with physical interpretations given.**

***Index Terms*—Cyber physical systems, intrusion detection, intrusion response, performance analysis.**

Acronyms

CPS	Cyber physical system
IDRS	Intrusion detection and response system
IDS	Intrusion detection system
RTU	Remote terminal unit
MTU	Master terminal unit
MTTF	Mean time to failure
SPN	Stochastic Petri net
SPNP	Stochastic Petri net package
XML	Extensible markup language
SNMP	Simple network management protocol
CDMA	Code division multiple access

Notation

N	Number of nodes
N_g	Number of good nodes
N_b	Number of bad nodes
N_b^a	Number of active bad nodes
N_b^i	Number of idle bad nodes
N_e	Number of evicted nodes
$\bar{n}$	Number of neighbors within radio range
α	α in $Beta(\alpha, \beta)$
β	β in $Beta(\alpha, \beta)$
γ	Number of ranging operations per node per T_{IDS}
E_t	Energy for transmission per node
E_r	Energy for reception per node
E_a	Energy for analyzing data per node
E_s	Energy for sensing per node
E_o	Initial system energy
T_{IDS}	Intrusion detection interval
E_{IDS}	Energy consumed per T_{IDS}
N_{IDS}	Maximum IDS cycles before energy exhaustion
m	Number of detectors used in the system IDS
X_b	Compliance degree of a bad node
X_g	Compliance degree of a good node
X_i	Compliance degree of arbitrary node i
c_i	ith compliance degree output
C_T	System minimum compliance threshold
C_T^p	Minimum threshold set by the system for the persistent attack case
δ_{C_T}	Increment to C_T per active bad node detected
$\hat{x}$	Estimate of x
p_{fn}	Per-node host IDS false negative probability
p_{fn}^p	Per-node host IDS false negative probability under persistent attacks
p_{fn}^r	Per-node host IDS false negative probability under random attacks
p_{fn}^i	Per-node host IDS false negative probability under insidious attacks
p_{fp}	Per-node host IDS false positive probability
p_{fp}^p	Per-node host IDS false positive probability under persistent attacks

Manuscript received October 23, 2011; revised May 01, 2012; accepted August 30, 2012. Date of publication January 29, 2013; date of current version February 27, 2013. This work was supported in part by the Army Research Office under Contracts W911NF-12-1-0016 and W911NF-12-1-0445. Associate Editor: S. Shieh.

The authors are with the Department of Computer Science, Virginia Polytechnic Institute and State University, Falls Church, VA 22043 USA (e-mail: rrmitche@vt.edu; irchen@vt.edu).

Digital Object Identifier 10.1109/TR.2013.2240891

p_{fp}^{r}	Per-node host IDS false positive probability under random attacks
p_{fp}^{i}	Per-node host IDS false positive probability under insidious attacks
$\mathcal{P}_{\text{fn}}$	System IDS false negative probability
$\mathcal{P}_{\text{fp}}$	System IDS false positive probability
p_{random}	Random attack probability by a random attacker
p_a	Attack probability
λ_c	Per-node capture rate
λ_{if}	Impairment rate for an attacker to cause severe functional impairment

I. Introduction

A cyber physical system (CPS) typically comprises sensors, actuators, control units, and physical objects for controlling and protecting a physical infrastructure. Because of the dire consequence of a CPS failure, protecting a CPS from malicious attacks is of paramount importance. In this paper, we address the reliability issue of a CPS designed to sustain malicious attacks over a prolonged mission period without energy replenishment. A CPS often operates in a rough environment wherein energy replenishment is not possible, and nodes may be compromised (or captured) at times. Thus, an intrusion detection and response system (IDRS) must detect malicious nodes without unnecessarily wasting energy to prolong the system lifetime.

Intrusion detection system (IDS) design for CPSs has attracted considerable attention [1], [7]. Detection techniques in general can be classified into three types: signature based, anomaly based, and specification based techniques. In the area of signature based IDS techniques, Oman and Phillips [22] study an IDS for CPSs that transforms data collected in extensible markup language (XML) format to Snort signatures in an electricity distribution laboratory. Verba and Milvich [26] study an IDS for CPSs that takes a multitrust hybrid approach using signature based detection and traffic analysis. Our work is different from these studies in that we use specification based detection rather than signature based detection to deal with unknown attacker patterns.

In the area of anomaly based IDS techniques, Barbosa and Pras [2] study an IDS for CPSs that tests state machine and Markov chain approaches to traffic analysis on a water distribution system based on a comprehensive vulnerability assessment. Linda *et al.* [18] study an IDS for CPSs that uses error-back propagation and Levenberg–Marquardt approaches with window based feature extraction. Gao *et al.* [16] study an IDS for CPSs that uses a three stage back propagation artificial neural network (ANN) based on Modbus features. Bellettini and Rrushi [4] study an IDS for CPSs that seeds the runtime stack with NULL calls, applies shuffle operations, and performs detection using product machines. Yang *et al.* [28] study an IDS for CPSs that uses the simple network management protocol (SNMP) to drive prediction, residual calculation, and detection modules for an experimental testbed. Bigham *et al.* [5] study an IDS for CPSs that demonstrates promising control of detection and false negative rates. Tsang and Kwong [25] study a rich multitrust IDS for CPSs that uses a novel machine learning approach. Xie *et al.* [27] survey anomaly detection techniques, and advocate an anomaly based layered approach. Our work is different from these studies in that we use specification based rather than anomaly based techniques to avoid using resource-constrained sensors or actuators in a CPS for profiling anomaly patterns (e.g., through learning), and to avoid high false positives (treating good nodes as bad nodes).

In the area of specification-based IDS techniques, Cheung *et al.* [12] study a specification based IDS that uses PVS to transform protocol, communication pattern, and service availability specifications into a format compatible with EMERALD. Carcano *et al.* [6] propose a specification based IDS that extends [15]; it distinguishes faults from attacks, describes a language to express a CPS specification, and establishes a critical state distance metric. Zimmer *et al.* [29] study a specification based IDS that instruments a target application, and uses a scheduler to confirm timing analysis results. Our work is also specification based. However, our work is different from these prior studies in that we automatically map a specification into a state machine consisting of good and bad states, and simply measure a node's deviation from good states at runtime for intrusion detection. Moreover we apply specification-based techniques to host-level intrusion detection only. To cope with incomplete, uncertain information available to nodes in the CPS, and to mitigate the effect of node collusion, we devise system-level intrusion detection based on multitrust to yield a low false alarm probability.

While the literature is abundant in the collection and analysis aspects of intrusion detection, the response aspect is little treated. In particular, there is a gap with respect to intrusion detection and response. Our IDRS design addresses both intrusion detection and response issues, with the goal to maximize the CPS lifetime.

Our methodology for CPS reliability assessment is model-based analysis. Specifically, we develop a probability model to assess the reliability property of a CPS equipped with an IDRS for detecting and responding to malicious events detected. Untreated in the literature, we consider a variety of attacker behaviors including persistent, random, and insidious attacker models, and identify the best design settings of the detection strength and response strength to best balance energy conservation versus intrusion tolerance for achieving high reliability, when given a set of parameter values characterizing the operational environment and network conditions. Parameterization of the model using the properties of the IDS system is one major contribution of the paper.

The rest of the paper is organized as follows. Section II gives the system model. Section III develops a mathematical model based on stochastic Petri nets [10], [11], [23] for theoretical analysis. Section IV discusses the parameterization process for the reference CPS. Section V presents numerical data with physical interpretations given. Finally, Section VI outlines some future research areas.

II. System Model

A. Reference CPS

Our reference CPS model is based on the CPS infrastructure described in [21] comprising 128 sensor carrying mobile nodes. Each node uses its sensor to measure any detectable phenomena nearby, and ranges its neighbors periodically by transmitting a code division multiple access (CDMA) waveform. Neighbors receiving that waveform transform the timing of the code (1023 symbols) and carrier (915 MHz) into distance. Essentially, each node performs sensing and reporting functions to provide information to upper layer control devices to control and protect the CPS infrastructure, and in addition utilizes its ranging function for node localization and intrusion detection.

The reference model is a special case of a single-enclave system with homogeneous nodes. The IDS functionality is distributed to all nodes in the system for intrusion and fault tolerance. On top of the sensor carrying mobile nodes sits an enclave control node responsible for setting system parameters in response to dynamically changing conditions such as changes of attacker strength. The control module is assumed to be fault and intrusion free through security and hardware protection mechanisms against capture attacks and hardware failure.

Fig. 1 contextualizes our reference CPS which comprises 128 sensor carrying mobile nodes, a control unit, and physical objects for controlling and protecting a physical infrastructure. The mobile nodes are capable of sensing physical environments, as well as actuating and controlling the underlying physical objects in the CPS. They function as sensors and actuators, each carrying sensors for sensing physical phenomena, as well as actuating devices for controlling physical objects. The CPS literature identifies these mobile nodes as remote terminal units (RTUs). Sitting on top of these mobile nodes is a control unit which receives sensing data from the mobile nodes and determines actions to be performed by individual nodes or a group of mobile nodes. The CPS literature identifies the control unit as the master terminal unit (MTU). The actions formulated by the MTU trigger actuating devices to control and protect the physical objects in the CPS. We exemplify a number of applications to which our reference CPS can apply.

1) Disaster recovery (say after an earthquake) might involve a group of mobile nodes with motion and video sensing and actuating capabilities cooperating under the control of a disaster corrective control unit to protect and recover physical objects (e.g., people or a physical infrastructure).
2) Emergency rescue (say a burning building) may require a group of mobile fighters equipped with motion and video sensing and fighting capabilities cooperating under the control of a control unit to rescue physical objects (e.g., people trapped or seized).
3) Military patrol (combat or reconnaissance) [13] might consist of a group of mobile patrol nodes equipped with motion sensing and fighting capabilities cooperating under the control of a control unit to protect and control physical objects (e.g., geographic areas or critical resources).
4) Pervasive healthcare [19] might use a group of mobile medical personnel equipped with motion and video sensing and actuating capabilities cooperating under the control of a control unit to protect and provide healthcare to physical objects (e.g., patients or medical devices).
5) Unmanned aircraft systems [20] might consist of a group of unmanned aerial vehicles equipped with sensing and aircraft fighting capabilities cooperating under the control of a remote control unit to control and protect physical objects (e.g., geographic areas).

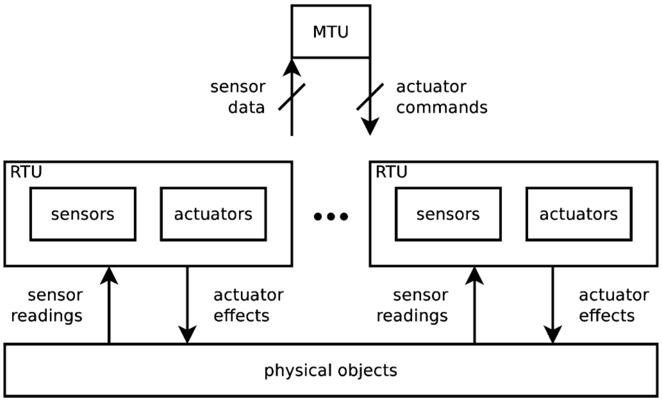

Fig. 1. Reference CPS.

Our reference CPS is distinct from Wireless Sensor Networks (WSNs). WSNs are resource constrained, mostly stationary, and have a specific traffic profile. On the other hand, our reference CPS is safety-critical, mobile, and uses ad hoc networking with bidirectional flows. We do not make any assumptions regarding the network structure used to connect nodes in a CPS. In our reference CPS, nodes are mobile, and they are connected through wireless links to the control node. Our host IDS design (Section II-D) is based on local monitoring, and our system-level IDS design (Section II-E) is based on the voting of neighbor monitoring nodes. Both IDS techniques can be generically applied to any network structure (such as a star configuration) used in a CPS.

B. Security Failure

While our approach is general enough to take any security failure definition, we consider two security failure conditions. The first condition is based on the Byzantine fault model [17]. That is, if one-third or more of the nodes are compromised, then the system fails. The reason is that once the system contains 1/3 or more compromised nodes, it is impossible to reach a consensus, hence inducing a security failure. The second condition is impairment failure. That is, a compromised CPS node performing active attacks without being detected can impair the functionality of the system and cause the system to fail. Impairment failure is modeled by defining an impairment-failure attack period by a compromised node beyond which the system cannot sustain the damage.

Specifically, a control unit in our reference CPS would take in multiple sensor readings (from sensor carrying mobile nodes) sensing the same physical phenomena to make a decision on actions to be performed by a set of actuators (also mobile nodes). The first failure mode, Byzantine failure, accounts for the condition that the control unit is not able to obtain any sensor reading consensus. The second failure mode, impairment failure, accounts for the condition that impairment by a bad node (especially an actuator) over an impairment-failure period without being detected will severely impair the system and cause the system to fail.

C. Attack Model

The first step in investigating network security is to define the attack model. We consider capture attacks which turn a good node into a bad insider node. At the sensor-actuator layer of the CPS architecture, a bad node can perform data spoofing attacks (reporting spoof sensor data) and bad command execution attacks. At the networking layer, a bad node can perform various communication attacks including selective forwarding, packet dropping, packet spoofing, packet replaying, packet flooding, and even Sybil attacks to disrupt the system's packet routing functionality. At the control layer, a bad node can perform control-level attacks including aggregated data spoofing attacks, and command spoofing attacks. Nodes at the control layer, however, are less susceptible to capture attacks because they are normally deployed in a physical confine which protects them from tampering. For this reason, in this paper, our primary interest is on capture attacks of sensor-actuator nodes performing basic sensing, actuating, and networking functions.

We consider three attacker models: persistent, random, and insidious. A persistent attacker performs attacks with probability one (i.e., whenever it has a chance). The primary objective is to cause impairment failure. A random attacker performs attacks randomly with probability p_{random}. The primary objective is to evade detection. It may take a longer time for a random attacker to cause impairment failure because the attack is random. However, random attackers are hidden so it may increase the probability of Byzantine security failure once the number of bad nodes equals or exceeds 1/3 of the node population. An insidious attacker is hidden all the time to evade detection until a critical mass of compromised nodes is reached to perform "all in" attacks. The primary objective is to maximize the failure probability caused by either impairment or Byzantine security failure.

D. Host Intrusion Detection

Our host intrusion detection protocol design is based on two core techniques: behavior rule specification, and vector similarity specification. The basic idea of behavior rule specification is to specify the behavior of an entity (a sensor or an actuator) by a set of rules from which a state machine is automatically derived. Then, node misbehavior can be assessed by observing the behaviors of the node against the state machine (or behavior rules). The basic idea of vector similarity specification is to compare similarity of a sequence of sensor readings, commands, or votes among entities performing the same set of functions. A state machine is also automatically derived from which a similarity test is performed to detect outliers. More specifically, the states derived in the state machine would be labeled as secure versus insecure. A monitoring node then applies snooping and overhearing techniques observing the percentage of time a neighbor node is in secure states over T_{IDS}. A longer sojourn time in secure states indicates greater specification compliance, while a shorter sojourn time indicates less specification compliance. If X_i falls below C_T, node i is considered compromised. We apply these two host IDS techniques to the reference CPS as follows. (a) A monitoring node periodically determines a sequence of locations of a sensor carrying mobile node within radio range through ranging, and detects if the location sequence (corresponding to the state sequence) deviates from the expected location sequence. (b) A monitoring node periodically collects votes from neighbor nodes who have participated in system intrusion detection (described below), and detects dissimilarity of vote sequences among these neighbors for outlier detection.

The measurement of compliance degree of a node frequently is not perfect, and can be affected by noise and unreliable wireless communication in the CPS. We model the compliance degree by a random variable X with $G(\cdot) = Beta(\alpha, \beta)$ distribution [24], with the value 0 indicating that the output is totally unacceptable (zero compliance), and 1 indicating the output is totally acceptable (perfect compliance), such that $G(a)$, $0 \leq a \leq 1$, is given by

$$G(a) = \int_0^a \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} dx, \tag{1}$$

and the expected value of X is given by

$$E_B[X] = \int_0^1 x \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} dx = \frac{\alpha}{\alpha+\beta.} \tag{2}$$

The α and β parameters are to be estimated based on the method of maximum likelihood by using the compliance degree history collected during the system's testing phase in which the system is tested with its anticipated attacker event profile, and where the compliance degree is assessed using the specification-based host IDS technique described earlier. A node's anticipated event profile describes a node's behaviors, and predicts the next state the node will be entering upon an event occurrence, given that the node is in its current state. For example, a persistent attacker will likely go to another bad state because it performs attacks continuously. A random attacker will likely go to a bad state in accordance to its random attack probability because it performs attacks randomly. A good node on the other hand will likely go to another good state because it complies with its behavior rules, unless the detection of its behaviors is hindered by noise or wireless channel error. The compliance degree history collected this way is the realization of a sequence of random variables $(c_1, c_2, \ldots, c_n)$, and n is the total number of compliance degree outputs observed. The maximum likelihood estimates of α and β are obtained by numerically solving

$$\begin{gathered}\frac{n\frac{\partial\Gamma(\hat{\alpha}+\hat{\beta})}{\partial\hat{\alpha}}}{\Gamma(\hat{\alpha}+\hat{\beta})} - \frac{n\frac{\partial\Gamma(\hat{\alpha})}{\partial\hat{\alpha}}}{\Gamma(\hat{\alpha})} + \sum_{i=1}^{n} \log c_i = 0 \\ \frac{n\frac{\partial\Gamma(\hat{\alpha}+\hat{\beta})}{\partial\hat{\beta}}}{\Gamma(\hat{\alpha}+\hat{\beta})} - \frac{n\frac{\partial\Gamma(\hat{\beta})}{\partial\hat{\beta}}}{\Gamma(\hat{\alpha})} + \sum_{i=1}^{n} \log(1-c_i) = 0\end{gathered} \tag{3}$$

where

$$\frac{\partial\Gamma(\hat{\alpha}+\hat{\beta})}{\partial\hat{\alpha}} = \int_0^\infty (\log x) x^{\hat{\alpha}+\hat{\beta}-1} e^{-x} dx.$$

A less general though simpler model is to consider a single parameter $Beta(\beta)$ distribution with α equal to 1. In this case, the density is $\beta(1-x)^{\beta-1}$ for $0 \leq x \leq 1$, and 0 otherwise. The maximum likelihood estimate of β is

$$\hat{\beta} = \frac{n}{\sum_{i=1}^{n} \log\left(\frac{1}{1-c_i}\right)} \tag{4}$$

Host intrusion detection is characterized by p_{fn} and p_{fp}. While many detection criteria [3], [8], [9] are possible, we consider a threshold criterion in this paper. That is, if X_b is higher than C_T, then there is a false negative. Suppose that X_b is modeled by a $G(\cdot) = Beta(\alpha, \beta)$ distribution as described above. Then p_{fn} is given by

$$p_{\mathrm{fn}} = \Pr\{X_b > C_T\} = 1 - G(C_T). \tag{5}$$

On the other hand, if X_g is less than C_T then there is a false positive. Again suppose that X_g is modeled by a $G(\cdot) = Beta(\alpha, \beta)$ distribution. Then p_{fp} is given by

$$p_{\mathrm{fp}} = \Pr\{X_g \leq C_T\} = G(C_T). \tag{6}$$

Here we observe that these two probabilities are largely affected by the setting of C_T. A large C_T induces a small false negative probability at the expense of a large false positive probability. Conversely, a small C_T induces a small false positive probability at the expense of a large false negative probability. A proper setting of C_T in response to attacker strength detected at runtime helps maximize the system lifetime.

E. System Intrusion Detection

Our system IDS technique is based on majority voting of host IDS results to cope with incomplete and uncertain information available to nodes in the CPS. Our system-level IDS technique involves the selection of m detectors as well as the invocation interval T_{IDS} to best balance energy conservation versus intrusion tolerance for achieving high reliability. Each node periodically exchanges its routing information, location, and identifier with its neighbor nodes. A coordinator is selected randomly among neighbors so that the adversaries will not have specific targets. We add randomness to the coordinator selection process by introducing a hashing function that takes in the identifier of a node concatenated with the current location of the node as the hash key. The node with the smallest returned hash value would then become the coordinator. Because candidate nodes know each other's identifier and location, they can, without trading information, execute the hash function to determine which node would be the coordinator. The coordinator then selects m detectors randomly (including itself), and lets all detectors know each others' identities so that each voter can send its yes or no vote to other detectors. Vote authenticity is achieved via preloaded public keys. At the end of the voting process, all detectors will know the same result; the node is diagnosed as good, or as bad based on the majority vote.

The system IDS is characterized by $\mathcal{P}_{\mathrm{fn}}$ and $\mathcal{P}_{\mathrm{fp}}$. These two false alarm probabilities are not constant but vary dynamically, depending on the percentage of bad nodes in the system when majority voting is performed. We will derive these two probabilities in the paper.

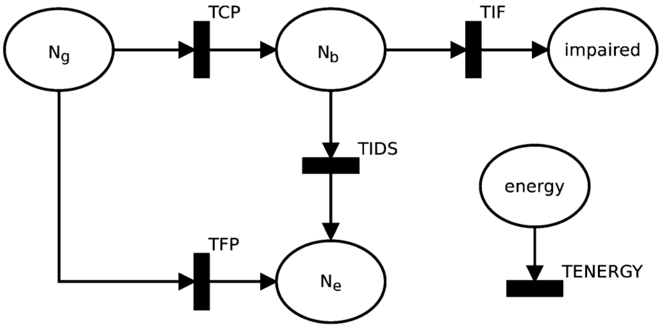

Fig. 2. SPN model for intrusion detection and response.

F. Intrusion Response

Our IDRS reacts to malicious events detected at runtime by adjusting C_T. For example, when it senses an increasing attacker strength, it can increase C_T with the objective to prevent impairment security failure. This approach results in a smaller false negative probability, which has a positive effect of reducing the number of bad nodes in the system, and decreasing the probability of impairment security failure. However, it also results in a larger false positive probability, which has the negative effect of reducing the number of good nodes in the system, and consequently increasing the probability of Byzantine security failure. To compensate for the negative effect, the IDRS increases the audit rate (by decreasing the intrusion detection interval) or increases the number of detectors to reduce the false positive probability at the expense of more energy consumption. The relationship between the minimum compliance threshold C_T set versus p_{fn} and p_{fp} must be determined at static time so the system can adjust C_T dynamically in response to malicious events detected at runtime.

III. Model and Analysis

Our theoretical model utilizes stochastic Petri net (SPN) techniques [14]. Fig. 2 shows the SPN model describing the ecosystem of a CPS with intrusion detection and response under capture, impairment, and Byzantine security attacks.

The underlying model of the SPN model is a continuous-time semi-Markov process with a state representation $(N_g, N_b, N_e, impaired, energy)$ where $impaired$ is a binary variable with 1 indicating impairment security failure, and $energy$ is a binary variable with 1 indicating energy availability and 0 indicating energy exhaustion.

The input parameters to the SPN model are N, λ_c, λ_{if}, p_{fp}, p_{fn}, T_{IDS}, m and p_{random}. The derived parameters are $\mathcal{P}_{\mathrm{fp}}$, $\mathcal{P}_{\mathrm{fn}}$, p_a, N_{IDS}. The only output parameter is MTTF.

Table I annotates transitions, and gives transition rates used in the SPN model. The SPN model shown in Fig. 2 is constructed as follows.

- We use tokens to represent nodes. We use places to hold tokens. Initially, all N nodes are good nodes (e.g., 128 in our reference CPS). Hence initially place N_g holds N tokens.
- We use transitions to model events. Specifically, TCP models good nodes being compromised; TFP models a

good node being falsely identified as compromised; TIDS models a bad node being detected correctly.

- Good nodes may become compromised because of capture attacks with rate λ_c. This assumption is modeled by associating transition TCP with an aggregate rate $\lambda_c \times N_g$. Firing TCP will move tokens one at a time (if it exists) from place N_g to place N_b. Tokens in place N_b represent bad nodes performing impairment attacks with probability p_a.
- When a bad node is detected by the system IDS as compromised, the number of compromised nodes evicted will be incremented by 1, so place N_e will hold one more token. On the other hand, the number of undetected compromised nodes will be decremented by 1, i.e., place N_b will hold one less token. These detection events are modeled by associating transition TIDS with a rate of $(N_b \times (1 - \mathcal{P}_{\text{fn}}))/T_{\text{IDS}}$, with $1 - \mathcal{P}_{\text{fn}}$ accounting for the system IDS true positive probability.
- The system-level IDS can incorrectly identify a good node as compromised. This is modeled by moving a good node in place N_g to place N_e from firing transition TFP with a rate of $(N_g \times \mathcal{P}_{\text{fp}})/T_{\text{IDS}}$, with $\mathcal{P}_{\text{fp}}$ accounting for the system IDS false positive probability.
- The system energy is exhausted after time $N_{\text{IDS}} \times T_{\text{IDS}}$, where N_{IDS} is the maximum number of intrusion detection intervals the CPS can possibly perform before it exhausts its energy due to performing ranging, sensing, and intrusion detection functions. It can be estimated by considering the amount of energy consumed in each T_{IDS} interval. This energy exhaustion event is modeled by placing a token in place $energy$ initially, and firing transition TENERGY with rate $1/(N_{\text{IDS}} \times T_{\text{IDS}})$. When the energy exhaustion event occurs, the token in place $energy$ will be vanished, and the system enters an absorbing state meaning the lifetime is over. This condition is modeled by disabling all transitions in the SPN model.
- When the number of bad nodes (i.e., tokens in place N_b) is at least 1/3 of the total number of nodes (tokens in places N_g and N_b), the system fails because of a Byzantine failure. The system lifetime is over, and is modeled again by disabling all transitions in the SPN model.
- Bad nodes in place N_b perform attacks with probability p_a, and cause impairment to the system. After an impairment-failure time period is elapsed, heavy impairment due to attacks will result in a security failure. We model this situation by firing transition TIF with a rate of $p_a \times N_b \times \lambda_{\text{if}}$ indicating the amount of time needed by $p_a N_b$ bad nodes to reach this level of impairment, beyond which the system cannot sustain the damage. The value of λ_{if} is system specific, and is determined by domain experts. A token is flown into place $impaired$ when such a security failure occurs. Once a token is in place $impaired$, the system enters an absorbing state, meaning the lifetime is over. Again, it is modeled by disabling all transitions in the SPN model.

Here we note that the last two bullet points cover the two conditions that would cause a security failure.

We utilize the SPN model to analyze two design tradeoffs.

- Detection strength versus energy consumption—As we increase the detection frequency (a smaller T_{IDS}) or the number of detectors (a larger m), the detection strength increases, thus preventing the system from running into a security failure. However, this increases the rate at which energy is consumed, thus resulting in a shorter lifetime. Consequently, there is an optimal setting of T_{IDS} and m under which the system MTTF is maximized, given the node capture rate and attack model.

TABLE I
TRANSITION RATES OF THE SPN MODEL

Transition Name	Rate
TENERGY	$\frac{1}{N_{\text{IDS}} \times T_{\text{IDS}}}$
TCP	$N_g \times \lambda_c$
TFP	$\frac{N_g \times \mathcal{P}_{\text{fp}}}{T_{\text{IDS}}}$
TIDS	$\frac{N_b \times (1 - \mathcal{P}_{\text{fn}})}{T_{\text{IDS}}}$
TIF	$p_a \times N_b \times \lambda_{\text{if}}$

- Detection response versus attacker strength—As the random attack probability p_a decreases, the attacker strength decreases, thus lowering the probability of security failure due to impairment attacks. However, compromised nodes become more hidden and difficult to detect because they leave less evidence traceable, resulting in a higher per-host false negative probability p_{fn}, and consequently a higher system-level false negative probability $\mathcal{P}_{\text{fn}}$. This increases the probability of security failure due to Byzantine attacks. The system can respond to a detected instantaneous attacker strength, and adjust C_T to trade a high per-host false positive probability p_{fp} for a low per-host false negative probability p_{fn}, or vice versa, so as to minimize the probability of security failure. Hence, there exists an optimal setting of C_T as a function of attacker strength detected at time t under which the system security failure probability is minimized.

Let L be a binary random variable denoting the lifetime of the system such that it takes on the value of 1 if the system is alive at time t, and 0 otherwise. Then, the expected value of L is the reliability of the system $R(t)$ at time t. Consequently, the integration of $R(t)$ from $t = 0$ to ∞ gives the mean time to failure (MTTF) or the average lifetime of the system we aim to maximize. The binary value assignment to L can be done by means of a reward function assigning a reward r_i of 0 or 1 to state i at time t as

$$r_i = \begin{cases} 1 & \text{if system is alive in state } i, \\ 0 & \text{if system fails due to security or energy failure.} \end{cases}$$

A state is represented by the distribution of tokens to places in the SPN model. For example, with the SPN model defined in Fig. 2, the underlying state is represented by $(N_g, N_b, N_e, impaired, energy)$. When place $energy$ contains zero tokens, it indicates energy exhaustion. When N_g is less than or equal to twice N_b, it indicates a Byzantine failure. When place $impaired$ contains a token, it indicates a security failure due to significant functional impairment. Once the binary value of 0 or 1 is assigned to all states of the system as described above, the reliability of the system $R(t)$ is the expected value of L weighted on the probability that the system stays at a particular state at time t, which we can obtain easily from solving the SPN model using stochastic Petri net

TABLE II
PARAMETERS AND THEIR VALUES FOR THE REFERENCE CPS

Parameter	Default value
N	128
$\bar{n}$	32
p_{fn}	[1-20%]
p_{fp}	[1-20%]
λ_c	1/[1-24hr]
λ_{if}	1/[12-48hr]
T_{IDS}	[1-60min]
m	[3,11]
γ	5
E_t	0.000125 J
E_r	0.00005 J
E_a	0.00174 J
E_s	0.0005 J
E_o	16128 kJ

package (SPNP) [14]. The MTTF of the system is equal to the cumulative reward to absorption, i.e.,

$$MTTF = \int_0^\infty R(t)dt, \tag{7}$$

which we can again compute easily using SPNP.

IV. Parameterization

We consider the reference CPS model introduced in Section II operating in a 2× 2 area with a network size (N) of 128 nodes initially. Hence, the number of neighbors within radio range $(\bar{n})$ initially is about $128/4 = 32 \text{ nodes}$. Our IDS design is based on local monitoring, so it can be generically applied to any network structure. A node in our reference CPS uses a 35 Wh battery, so its energy is 126000 J. The system energy initially (E_o) is therefore 126000 J × 128 = 16128000 J. Table II lists the set of parameters and their values for the reference CPS.

A. System-Level IDS $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$

We first parameterize the system IDS $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$ given per-host IDS false positive probability p_{fp} and per-host IDS false negative probability p_{fn} as input. We first note that $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$ highly depend on the attacker behavior. A persistent attacker constantly performs slandering attacks such that it will vote a bad node as a good node, and conversely a good node as a bad node, to eventually cause a security failure. However, a random or an insidious attacker will only perform slandering attacks randomly with probability p_a to avoid detection.

We first differentiate the number of active bad nodes, N_b^a, from the number of inactive bad nodes, N_b^i, with $N_b^a + N_b^i = N_b$, such that at any time

$$N_b^a = p_a \times N_b \tag{8}$$

$$N_b^i = (1 - p_a) \times N_b \tag{9}$$

The difference between an active bad node and an inactive bad node is that an inactive bad node behaves as if it were a good node to evade detection, including casting votes the same way as a good node would, when it participates in the system-level IDS voting process.

For a persistent attacker, $p_a = 1$. For a random attacker, $p_a = p_{\text{random}}$. For an insidious attacker, to maximize the benefit of colluding attacks, a compromised node stays dormant until a critical mass of compromised nodes is gathered so that $p_a = 1$ when $N_b \geq N_b^T$, and $p_a = 0$ otherwise, where N_b^T is a parameter reflecting the insidiousness degree. In other words, all bad nodes engage in active attacks when there is a critical mass of compromised nodes in the system.

We calculate $\mathcal{P}_{\text{fn}}$ by (10) (See equation at bottom of page). The equation for $\mathcal{P}_{\text{fp}}^p$ is the same except replacing p_{fn} by p_{fp} in the right hand side expression.

We explain (10) for obtaining $\mathcal{P}_{\text{fn}}$ in detail below. The explanation for $\mathcal{P}_{\text{fp}}$ follows the same logic. In (10), m this is the number of detectors, and m_a is the majority of m. The first summation aggregates the probability of a false negative stemming from selecting a majority of active bad nodes. That is, it is equal to the number of ways to choose a majority of m nodes from the set of active bad nodes times the number of ways to choose a minority of m nodes from the set of good nodes, and inactive bad nodes divided by the number of ways to choose m nodes from the set of all good and bad nodes. The second summation aggregates the probability of a false negative stemming from selecting a minority of m nodes from the set of active bad nodes which always cast incorrect votes, coupled with selecting a sufficient number of nodes from the set of good nodes and inactive bad nodes which make incorrect votes with probability p_{fn}, resulting in a majority of incorrect votes being cast.

B. Host IDS p_{fn} and p_{fp}

Next, we parameterize the host IDS false negative probability p_{fn} and false positive probability p_{fp} for persistent, random, and insidious attacks. For the case of persistent attacks with $p_a = 1$, the system, after a thorough testing and debugging phase, determines a minimum threshold C_T^p such that p_{fn}^p and p_{fp}^p, measured respectively based on (5) and (6), are acceptable to system design.

For the case of random attacks with probability $p_a < 1$, conceivably the amount of evidence observable from a bad node would be diminished proportional to p_a. Consequently, with the same minimum threshold C_T^p being used, the host false negative probability would increase. We again utilize (5), and (6) to respectively obtain p_{fn}^r, and p_{fp}^r for each given p_a value during the testing and debugging phase. Here we note that the host false

$$\mathcal{P}_{\text{fn}} = \sum_{i=0}^{m-m_a} \left[\frac{\binom{N_b^a}{m_a+i}\binom{N_g+N_b^i}{m-(m_a+i)}}{\binom{N_g+N_b^a+N_b^i}{m}} \right] + \sum_{j=0}^{m-m_a} \left[\frac{\binom{N_b^a}{j} \sum_{k=m_a-j}^{m-j} \left[\binom{N_g+N_b^i}{k} (p_{\text{fn}})^k \binom{N_g+N_b^i-k}{m-j-k} (1-p_{\text{fn}})^{(m-j-k)} \right]}{\binom{N_g+N_b^i+N_b^a}{m}} \right]. \tag{10}$$

TABLE III
β IN BETA(1,β) AND RESULTING p_{fn}, AND p_{fp} VALUES UNDER VARIOUS ATTACK MODELS

Attack Type	β	p_{fn}	p_{fp}
Random with P_a=1.000 (Persistent)	1.20	6.3%	7.3%
Random with P_a=0.800	1.00	10.0%	7.3%
Random with P_a=0.400	0.75	17.8%	7.3%
Random with P_a=0.200	0.50	31.6%	7.3%
Random with P_a=0.100	0.20	63.1%	7.3%
Random with P_a=0.050	0.13	74.1%	7.3%
Random with P_a=0.025	0.09	81.3%	7.3%
Insidious	0; 1.20	100%; 6.3%	7.3%

positive probability would remain the same, i.e. $p^r_{\text{fp}} = p^p_{\text{fp}}$, because the attacker behavior does not affect false positives, given the same minimum threshold C^p_T being used.

Lastly, under insidious attacks, the false positive probability is not affected, so $p^i_{\text{fp}} = p^p_{\text{fp}}$. Because insidious nodes stay dormant until a critical mass is achieved to perform "all in" attacks, the false negative probability is one during the dormant period, and is equal to that under persistent attacks during the "all in" attack period. Specifically,

$$p^i_{\text{fn}} = \begin{cases} p^p_{\text{fn}} & \text{if } N_b \geq N^T_b, \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

Here we note that p_{fn} and p_{fp} obtained above for persistent, random, or insidious attacks would be a function of time as input to (10) for calculating system-level IDS $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$ dynamically.

We apply the statistical analysis described by (1)–(4) to get the maximum likelihood estimates of β (with α set as 1) under each attacker behavior model, and then utilize (5) and (6) to yield p_{fn} and p_{fp}. The system minimum threshold C_T is set to $C^p_T = 0.9$ to yield $p^p_{\text{fn}} = 6.3\%$, and $p^p_{\text{fp}} = 7.3\%$. Table III summarizes β values, and the resulting p_{fn} and p_{fp} values under various attacker behavior models. The persistent attack model is a special case in which $p_a = 1$. The insidious attack model is another special case in which $p_a = 1$ during the "all in" attack period, and $p_a = 0$ during the dormant period.

C. Parameterizing C_T for Dynamic Intrusion Response

The parameterization of p_{fn} and p_{fp} above is based on a constant C_T being used (i.e., $C^p_T = 0.9$). A dynamic IDS response design is to adjust C_T in response to the attacker strength detected with the goal to maximize the system lifetime. The attacker strength of a node, say node i, may be estimated periodically by node i's intrusion detectors. That is, the compliance degree value of node i, $X_i(t)$, as collected by m intrusion detectors based on observations collected during $[t - T_{\text{IDS}}, t]$, is compared against the minimum threshold C^p_T set for persistent attacks. If $X_i(t) < C^p_T$, then node i is considered a bad node performing active attacks at time t; otherwise, it is a good node. This information is passed to the control module who subsequently estimates $N^a_b(t)$, representing the attacker strength at time t.

In this paper, we investigate a simple yet efficient IDS response design. The basic idea is to decrease the per-host false negative probability p_{fn} when the attacker strength is high, so we may quickly remove active attackers from the system to prevent impairment failure. This goal is achieved by increasing the C_T value. Conversely, when there is little attacker evidence detected, we lower C_T so we may quickly decrease the probability of a good node being misidentified as a bad node, i.e., lowering the per-host false positive probability, to prevent Byzantine failure.

While there are many possible ways to dynamically control C_T, in this paper we consider a linear one-to-one mapping function as

$$C_T(t) = C^p_T + \delta_{C_T} \times (N^a_b - 1) \tag{12}$$

Essentially we set C_T to C^p_T when $N^a_b(t)$ detected at time t is 1, and linearly increase (or decrease) C_T with increasing (or decreasing) attacker strength detected. With $C^p_T = 0.9$ in our CPS reference system, we set $\delta_{C_T} = 0.5$ and parameterize $C_T(t)$ as

$$C_T(t) = \begin{cases} 0.85 & \text{if } N^a_b = 0 \\ 0.90 & \text{if } N^a_b = 1 \\ 0.95 & \text{if } N^a_b = 2 \\ 0.99 & \text{if } N^a_b \geq 3 \end{cases} \tag{13}$$

Note that when C_T is closer to 1, a node will more likely be considered as compromised even if it wanders only for a small amount of time in insecure states. A large C_T induces a small per-host false negative probability p_{fn} at the expense of a large per-host false positive probability p_{fp}.

D. Energy

Lastly, we parameterize N_{IDS}, the maximum number of intrusion detection cycles the system can possibly perform before energy exhaustion, as

$$N_{\text{IDS}} = \frac{E_o}{E_{\text{IDS}}} \tag{14}$$

where E_{IDS} is the energy consumed per T_{IDS} interval due to ranging, sensing, and intrusion detection functions, calculated as

$$E_{\text{IDS}} = N \times (E_{\text{ranging}} + E_{\text{sensing}} + E_{\text{detection}}) \tag{15}$$

where E_{ranging}, E_{sensing}, $E_{\text{detection}}$ stand for energy spent for ranging, sensing, and intrusion detection in a T_{IDS} interval, respectively. Here the energy spent per node is multiplied with the node population in the CPS to get the total energy spent by all nodes per cycle.

In (15), E_{ranging} stands for the energy spent for periodic ranging. It is calculated as

$$E_{\text{ranging}} = \gamma \times [E_t + \bar{n} \times (E_r + E_a)] \tag{16}$$

Here a node spends E_t energy to transmit a CDMA waveform. Its $\bar{n}$ neighbors each spend E_r energy to receive the waveform, and each spend E_a energy to transform it into distance. This operation is repeated for γ times for determining a sequence of locations. In (15), E_{sensing} stands for the amount of energy consumed due to periodic sensing. It is computed as

$$E_{\text{sensing}} = \bar{n} \times (E_s + E_a). \tag{17}$$

Here a node spends E_s energy for sensing navigation and multipath mitigation data, and E_a energy for analyzing sensed data for each of its $\bar{n}$ neighbors. Finally, $E_{\text{detection}}$ stands for the energy used for performing intrusion detection on a target node. It can be calculated by

$$E_{\text{detection}} = m \times (E_t + \bar{n} \cdot E_r) + m \times (E_t + (m-1) \cdot (E_r + E_a)) . \tag{18}$$

Here we consider the energy required to choose m intrusion detectors to evaluate a target node (the first term), and the energy required for m intrusion detectors to vote (the second term). Specifically, the first term is the number of intrusion detectors times the cost of transmitting plus the number of nodes in radio range times the cost of receiving. The second term is the number of intrusion detectors times the cost of transmitting plus the number of peer intrusion detectors times the cost of receiving plus the cost of analyzing the vote.

V. Numerical Data

In this section, we present numerical data for reliability assessment as a result of executing intrusion detection and response in a CPS. Our objective is to identify optimal design settings in terms of the optimal values of T_{IDS}, m, and C_T under which we can best trade energy consumption versus intrusion detection, as well as response effectiveness versus impairment security failure, to maximize the system MTTF, when given a set of parameter values characterizing the operational and networking conditions.

A. Effect of Intrusion Detection Strength

We first examine the effect of intrusion detection strength measured by the intrusion interval, T_{IDS}, and the number of intrusion detectors, m. We only present results for the reference CPS under persistent attacks, as the results for other types of attacks show similar trends.

Fig. 3 shows MTTF versus T_{IDS} as the number of detectors (m) in the system-level IDS varies over the range of [3,11] in increments of 2. We see that there exists an optimal T_{IDS} value at which the system lifetime is maximized to best tradeoff energy consumption versus intrusion tolerance. Initially, when T_{IDS} is too small, the system performs ranging, sensing, and intrusion detection too frequently, and quickly exhausts its energy, resulting in a small lifetime. As T_{IDS} increases, the system saves more energy, and its lifetime increases. Finally, when T_{IDS} is too large, although the system can save even more energy, it fails to catch bad nodes often enough, resulting in the system having many bad nodes. Bad nodes through active attacks can cause impairment security failure. Furthermore, when the system has 1/3 or more bad nodes out of the total population, a Byzantine failure ensues. We observe that the optimal T_{IDS} value at which the system MTTF is maximized is sensitive to the m value. The general trend is that, as m increases, the optimal T_{IDS} value decreases. Here we observe that $m = 7$ is optimal to yield the maximum MTTF because too many intrusion detectors would induce energy exhaustion failure, while too few intrusion detectors would induce security failure. Using $m = 7$ can best balance energy exhaustion failure versus security failure for high reliability.

Fig. 4 shows MTTF versus T_{IDS} as the compromising rate λ_c varies over the range of once per 4 hours to once per 24 hours to test the sensitivity of MTTF with respect to λ_c (with m fixed at five to isolate its effect). We first observe that, as λ_c increases, MTTF decreases because a higher λ_c will cause more compromised nodes to be present in the system. We also observe that the optimal T_{IDS} decreases as λ_c increases. This happens because, when more compromised nodes exist, the system needs to execute intrusion detection more frequently to maximize MTTF. Fig. 4 identifies the best T_{IDS} to be used to maximize the lifetime of the reference CPS to balance energy exhaustion versus security failure, when given C_T and λ_c characterizing the operational and networking conditions of the system.

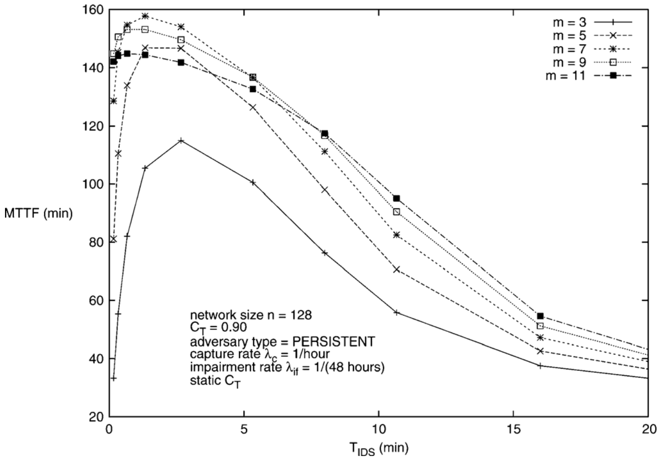

Fig. 3. MTTF versus T_{IDS} and m.

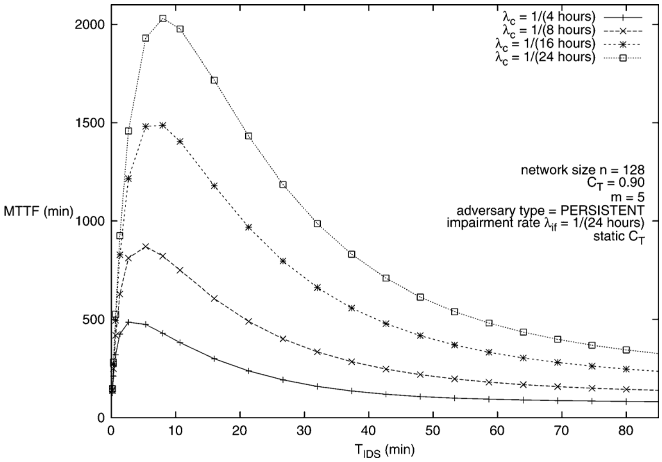

Fig. 4. MTTF versus T_{IDS} and λ_c.

B. Effect of Attacker Behavior

In this section, we analyze the effect of various attacker behavior models, including persistent (with $p_a = 1$, p^i_{fn} and p^i_{fp} given as input), random (with $p_a = p_{\text{random}}$, p^r_{fn}, and p^r_{fp} given as input), and insidious attacks (with $p_a = 1$ when $N_b \geq N^T_b = 10$ and $p_a = 0$ otherwise, p^i_{fp}, and p^i_{fn} defined by (11) given as input). The analysis conducted here is based on static C_T. In the next section, we will analyze the effect of dynamic C_T as a response to attacker strength detected at runtime.

Fig. 5 shows MTTF versus T_{IDS} with varying p_{random} values. We first observe that the system MTTF is low when p_{random}

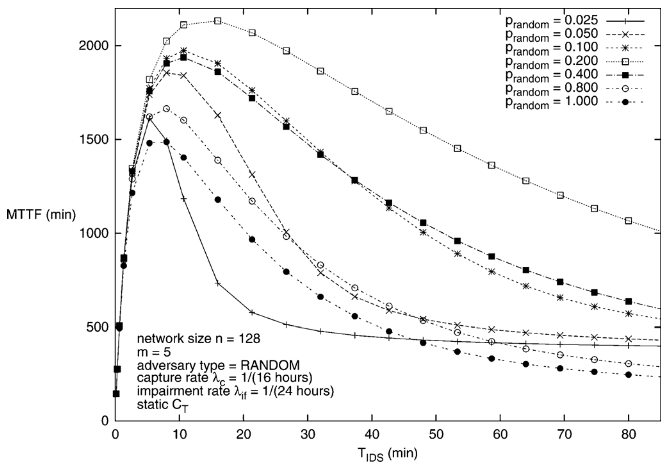

Fig. 5. MTTF versus $T_{\rm IDS}$ and $p_{\rm random}$.

is small (e.g., $p_{\rm random} = 0.025$). This happens because, when $p_{\rm random}$ is small, most bad nodes are dormant and remain in the system without being detected. Thus, the system suffers from Byzantine failure quickly, leading to a low MTTF. As $p_{\rm random}$ increases from 0.025 to 0.2, the system MTTF increases because of a higher chance of bad nodes being detected and removed from the system, thus reducing the probability of Byzantine security failure. As $p_{\rm random}$ increases further, however, the system MTTF decreases again because of a higher probability of impairment security failure as there will be more bad nodes actively performing impairment attacks. In the extreme case of $p_{\rm random} = 1$, all bad nodes perform attacks, and the system failure is mainly caused by impairment. The maximum MTTF occurs when $p_{\rm random} = 0.2$, at which point the probability of security failure due to either type of security attacks is minimized. Here we should note that the result of $p_{\rm random} = 0.2$ yielding the highest MTTF is a balance of impairment security failure rate versus Byzantine failure rate dictated by the parameter settings of the reference CPS as given in Tables II and III.

Fig. 6 compares the MTTF versus $T_{\rm IDS}$ of the reference CPS under the three attacker types head-to-head. It shows that the MTTF is the highest for the reference CPS under random attacks. The MTTF of the CPS under persistent attacks is the second highest. As expected, the reference CPS under insidious attacks has the lowest MTTF. We attribute this result to the fact that insidious attacks while dormant can cause Byzantine failure, and while “all in” can cause impairment failure. The extent to which the system MTTF differs depends on the relative rate at which impairment failure versus Byzantine failure occurs. The former is dictated by $\lambda_{\rm if}$, and the latter is dictated by how fast the Byzantine failure condition is satisfied. The result that the MTTF difference between persistent attacks (the second curve) and insidious attacks (the last curve) is relatively significant is due to a large Byzantine failure rate compared with the impairment failure rate. On the other hand, the reference CPS under random attacks can more effectively prevent either Byzantine failure or impairment failure from occurring by removing bad nodes as soon as they perform attacks. The system MTTF difference between random versus persistent attacks again depends on the relative rate at which impairment failure versus Byzantine failure occurs.

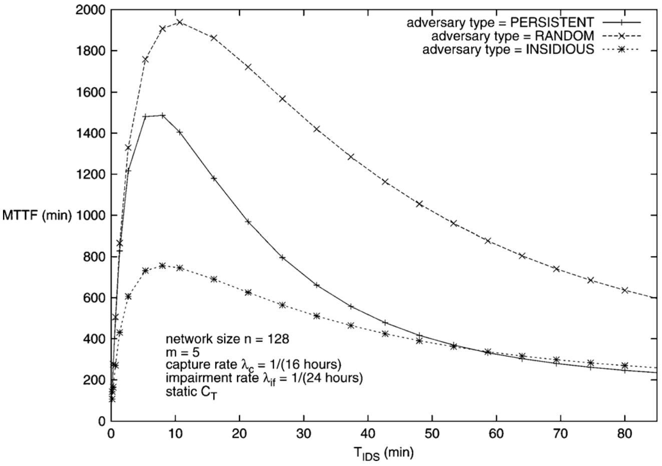

Fig. 6. MTTF versus $T_{\rm IDS}$ and adversary type.

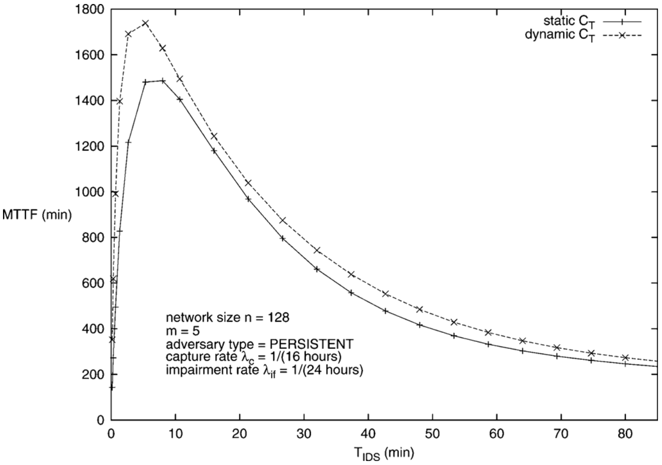

Fig. 7. Static versus dynamic response under persistent attacks.

C. Effect of Intrusion Response

In this section, we analyze the effect of intrusion response, i.e., dynamic C_T as a response to attacker strength detected at runtime, on the system MTTF.

Fig. 7 shows MTTF versus $T_{\rm IDS}$ under the static C_T design and the dynamic C_T design for the persistent attack case. We see there is a significant gain in MTTF under dynamic C_T over static C_T. The reason is that, with persistent attacks, all bad nodes are actively performing attacks, so the system is better off by increasing C_T to a high level to quickly remove bad nodes to prevent impairment failure. We observe that the optimal $T_{\rm IDS}$ value under the dynamic C_T design is smaller than that under the static C_T design to quickly remove nodes that perform active attacks.

Fig. 8 shows the MTTF versus $T_{\rm IDS}$ under the static C_T design and the dynamic C_T design for the random attack case with $p_{\rm random} = 0.2$. We pick the case of $p_{\rm random} = 0.2$ because it yields the highest MTTF among all random attack cases in the

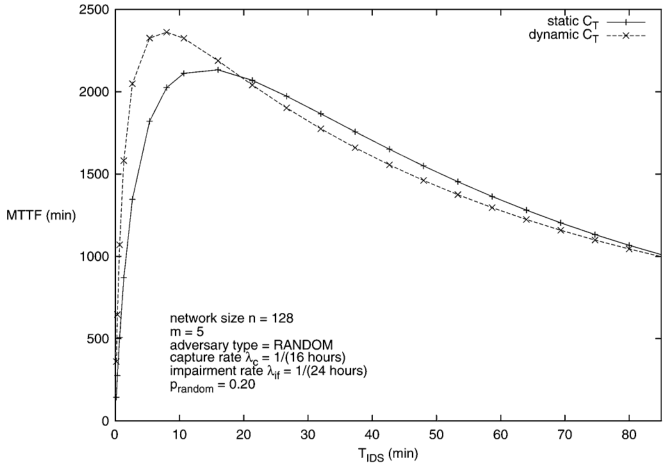

Fig. 8. Static versus dynamic response under random attacks.

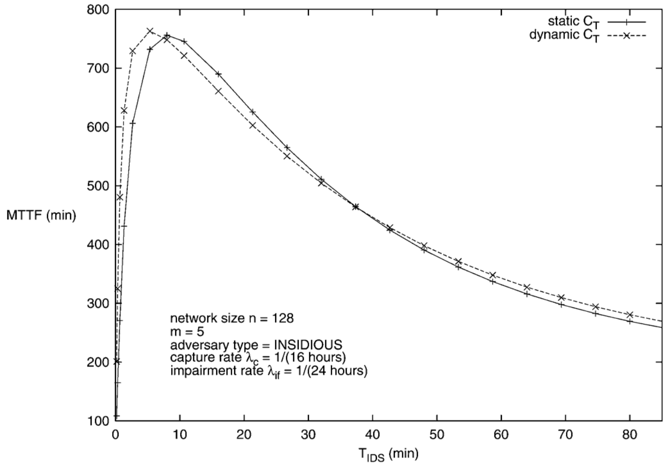

Fig. 9. Static versus dynamic response under insidious attacks.

reference CPS system (see Fig. 5). Here again we observe that dynamic C_T performs significantly better than static C_T, when operating at the identified optimal T_{IDS} value. The optimal T_{IDS} value under dynamic C_T again is smaller than that under static C_T to quickly remove nodes that perform active attacks.

Fig. 9 shows the MTTF versus T_{IDS} under the static C_T design and the dynamic C_T design for the insidious attack case. Here we observe the MTTF difference is relatively small compared with persistent or random attacks. The reason is that bad nodes do not perform active attacks until a critical mass is reached, so dynamic C_T would set a lower C_T value during the dormant period while rapidly setting a higher C_T value during the attack period. Because the attack period is relatively short compared with the dormant period, the gain in MTTF isn't very significant. Nevertheless, we observe, even for insidious attacks, dynamic C_T still performs better than static C_T.

As our C_T dynamic control function (12) adjusts C_T solely based on the attacker strength detected regardless of the attacker type, we conclude that the dynamic C_T design as a response to attacker strength detected at runtime can improve MTTF compared with the static C_T design.

VI. Future Work

In this paper, we developed a probability model to analyze the reliability of a cyber physical system in the presence of both malicious nodes exhibiting a range of attacker behaviors, and an intrusion detection and response system for detecting and responding to malicious events at runtime. For each attacker behavior, we identified the best detection strength (in terms of the detection interval and the number of detectors), and the best response strength (in terms of the per-host minimum compliance threshold for setting the false positive and negative probabilities), under which the reliability of the system may be maximized.

There are several future research directions, including (a) investigating other intrusion detection criteria (e.g., based on accumulation of deviation from good states) other than the current binary criterion used in the paper based on a minimum compliance threshold to improve the false negative probability without compromising the false positive probability; (b) investigating other intrusion response criteria (e.g., exponential increase of the minimum compliance threshold) other than the linear function used in the paper, and analyzing the effect on the system lifetime; (c) exploring other attack behavior models (e.g., an oracle attacker that can adjust the attacker strength depending on the detection strength to maximize security failure), and investigating the best dynamic response design to cope with such attacks; (d) developing a more elaborate model to describe the relationship between intrusion responses and attacker behaviors, and justifying such a relationship model by means of extensive empirical studies; and (e) extending the analysis to hierarchically-structured intrusion detection and response system design for a large CPS consisting of multiple enclaves each comprising heterogeneous entities subject to different operational and environment conditions and attack threats.

References

[1] M. Anand, E. Cronin, M. Sherr, M. Blaze, Z. Ives, and I. Lee, "Security challenges in next generation cyber physical systems," in *Beyond SCADA: Networked Embedded Control for Cyber Physical Systems*. Pittsburgh, PA, USA: NSF TRUST Science and Technology Center, Nov. 2006.

[2] R. Barbosa and A. Pras, "Intrusion detection in SCADA networks," in *Mechanisms for Autonomous Management of Networks and Services*, B. Stiller and F. De Turck, Eds. Berlin, Germany: Springer-Verlag, 2010, vol. 6155, Lecture Notes in Computer Science, pp. 163–166.

[3] F. B. Bastani, I. R. Chen, and T. W. Tsao, "Reliability of systems with fuzzy-failure criterion," in *Proc. Annu. Rel. Maintainability Symp.*, Anaheim, California, USA, January 1994, pp. 442–448.

[4] C. Bellettini and J. Rrushi, "A product machine model for anomaly detection of interposition attacks on cyber-physical systems," in *Proc. 23rd IFIP Int. Inf. Security Conf.*, Milan, Italy, Sep. 2008, pp. 285–300.

[5] J. Bigham, D. Gamez, and N. Lu, "Safeguarding SCADA systems with anomaly detection," in *Computer Network Security*, V. Gorodetsky, L. Popyack, and V. Skormin, Eds. Berlin, Germany: Springer-Verlag, 2003, vol. 2776, Lecture Notes in Computer Science, pp. 171–182.

[6] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Fovino, and A. Trombetta, "A multidimensional critical state analysis for detecting intrusions in SCADA systems," *IEEE Trans. Ind. Inf.*, vol. 7, no. 2, pp. 179–186, May 2011.

[7] A. Cárdenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," in *Proc. 1st Workshop Cyber-Phys. Syst.Security*, Virginia, USA, Jul. 2009.

[8] I. R. Chen and F. B. Bastani, "Effect of artificial-intelligence planning-procedures on system reliability," *IEEE Trans. Reliability*, vol. 40, no. 3, pp. 364–369, Sep. 1991.

[9] I. R. Chen, F. B. Bastani, and T. W. Tsao, "On the reliability of AI planning software in real-time applications," *IEEE Trans. Knowledge Data Eng.* , vol. 7, no. 1, pp. 4–13, 1995.

[10] I. R. Chen and D. C. Wang, "Analysis of replicated data with repair dependency," *Comput. J.*, vol. 39, no. 9, pp. 767–779, 1996.

[11] I. R. Chen and D. C. Wang, "Analyzing dynamic voting using Petri nets," in *Proc. 15th IEEE Symp. Rel. Distrib. Syst.*, Niagara Falls, Canada, Oct. 1996, pp. 44–53.

[12] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proc. SCADA Security Sci. Symp.*, Miami, FL, USA, Jan. 2007, pp. 127–134.

[13] J. H. Cho, I. R. Chen, and P. G. Feng, "Effect of intrusion detection on reliability of mission-oriented mobile group systems in mobile ad hoc networks," *IEEE Trans. Reliability*, vol. 59, no. 1, pp. 231–241, Mar. 2010.

[14] G. Ciardo, J. Muppala, and K. Trivedi, "SPNP: Stochastic Petri net package," in *Proc. 3rd Int. Workshop Petri Nets Perform. Models*, Washington, DC, USA, December 1989, pp. 142–151.

[15] I. Fovino, A. Carcano, T. De Lacheze Murel, A. Trombetta, and M. Masera, "Modbus/DNP3 state-based intrusion detection system," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Perth, Australia, April 2010, pp. 729–736.

[16] W. Gao, T. Morris, B. Reaves, and D. Richey, "On SCADA control system command and response injection and intrusion detection," in *Proc. 5th Annu. APWG eCrime Res. Summit (eCrime)*, Dallas, TX, USA, October 2010, pp. 1–9.

[17] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.

[18] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *Proc. Int. Joint Conf. Neural Netw.*, Atlanta, GA, USA, June 2009, pp. 1827–1834.

[19] R. Mitchell and I. R. Chen, "Behavior rule based intrusion detection for supporting secure medical cyber physical systems," in *Proc. IEEE Int. Conf. on Comput. Commun. ication Netw.*, Munich, Germany, Jul. 2012.

[20] R. Mitchell and I. R. Chen, "Specification based intrusion detection for unmanned aircraft systems," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Hilton Head Island, SC, USA, Jun. 2012, pp. 31–36.

[21] U.S. Department of Homeland Security, BAA-09-02 Geospatial Location Accountability and Navigation System for Emergency Responders (GLANSER) 2009.

[22] P. Oman and M. Phillips, "Intrusion detection and event monitoring in SCADA networks," in *Critical Infrastructure Protection*, E. Goetz and S. Shenoi, Eds. Boston: Springer, 2007, vol. 253, International Federation for Information Processing, pp. 161–173.

[23] A. P. Robin, A. Sahner, and K. S. Trivedi, *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package*. New York, NY, USA: Kluwer Academic Publishers, 2002.

[24] S. M. Ross, *Introduction to Probability Models*, 10th ed. New York, NY, USA: Academic Press, 2009.

[25] C.-H. Tsang and S. Kwong, "Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction," in *Proc. IEEE Int. Conf. Ind. Technol.*, Hong Kong, Dec. 2005, pp. 51–56.

[26] J. Verba and M. Milvich, "Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS)," in *Proc. IEEE Conf. on Technol. Homeland Security*, Idaho Falls, USA, May 2008, pp. 469–473.

[27] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1302–1325, July 2011.

[28] D. Yang, A. Usynin, and J. Hines, "Anomaly-based intrusion detection for SCADA systems," in *Proc. 5th Int. Topical Meeting Nuclear Plant Instrum., Contr. Human Mach. Interface Technol.*, Albuquerque, NM, USA, Nov. 2005, pp. 12–16.

[29] C. Zimmer, B. Bhat, F. Mueller, and S. Mohan, "Time-based intrusion detection in cyber-physical systems," in *Proc. 1st ACM/IEEE Int. Conf. Cyber-Physical Syst.* , Stockholm, Sweden, 2010, pp. 109–118.

Robert Mitchell received the B.S. and M.S. degrees in computer science from Virginia Polytechnic Institute, and State University, Blacksburg, USA, in 1997, and 1998, respectively. He is currently working toward the Ph.D. degree in the Department of Computer Science at Virginia Tech, Blacksburg, USA.

His research interests include security, mobile computing, sensor networks, embedded systems, and coding and information theory.

Ing-Ray Chen (M'10) received the B.S. degree from the National Taiwan University, Taipei, Taiwan; and the M.S. and Ph.D. degrees in computer science from the University of Houston, Houston, TX, USA.

He is currently a Professor in the Department of Computer Science at Virginia Tech, Blacksburg, USA. His research interests include mobile computing, wireless networks, security, intrusion detection, trust management, real-time intelligent systems, and reliability and performance analysis.

Dr. Chen currently serves as an Editor for IEEE COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, *Wireless Communications and Mobile Computing*, *The Computer Journal*, *Wireless Personal Communications*, and *Security and Communication Networks*. He is a Member of ACM.