IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING VOL : PP NO : 99 YEAR 2014

# Behavior Rule Specification-based Intrusion Detection for Safety Critical Medical Cyber Physical Systems

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

***Abstract*—We propose and analyze a behavior-rule specification-based technique for intrusion detection of medical devices embedded in a medical cyber physical system (MCPS) in which the patient's safety is of the utmost importance. We propose a methodology to transform behavior rules to a state machine, so that a device that is being monitored for its behavior can easily be checked against the transformed state machine for deviation from its behavior specification. Using vital sign monitor medical devices as an example, we demonstrate that our intrusion detection technique can effectively trade false positives off for a high detection probability to cope with more sophisticated and hidden attackers to support ultra safe and secure MCPS applications. Moreover, through a comparative analysis, we demonstrate that our behavior-rule specification-based IDS technique outperforms two existing anomaly-based techniques for detecting abnormal patient behaviors in pervasive healthcare applications.**

**keywords:** intrusion detection, sensor actuator networks, medical cyber physical systems, healthcare, security, safety.

## I. Introduction

The most prominent characteristic of a medical cyber physical system (MCPS) is its feedback loop that acts on the physical environment. In other words, the physical environment provides data to the MCPS sensors whose data feed the MCPS control algorithms that drive the actuators which change the physical environment. MCPSs are often characterized by sophisticated patient treatment algorithms interacting with the physical environment including the patient. In this paper, we are concerned with intrusion detection mechanisms for detecting compromised sensors or actuators embedded in an MCPS for supporting safe and secure MCPS applications upon which patients and healthcare personnel can depend with high confidence.

Intrusion detection system (IDS) design for cyber physical systems (CPSs) has attracted considerable attention [3], [7] because of the dire consequence of CPS failure. However, IDS techniques for MCPSs is still in its infancy with very little work reported. Intrusion detection techniques in general can be classified into four types: signature, anomaly, trust, and specification-based techniques. In this paper, we consider specification rather than signature-based detection to deal with unknown attacker patterns. We consider specification rather than anomaly based techniques to avoid using resource-constrained sensors or actuators in an MCPS for profiling anomaly patterns (e.g., through learning) and to avoid high false positives. We consider specification rather than trust-based techniques [5] to avoid delay due to trust aggregation and propagation to promptly react to malicious behaviors in safety critical MCPSs.

To accommodate resource-constrained sensors and actuators in an MCPS, we propose behavior-rule specification-based intrusion detection (BSID) which uses the notion of behavior rules for specifying acceptable behaviors of medical devices in an MCPS. Rule-based intrusion detection thus far has been applied only in the context of communication networks which have no concern of physical environments and the closed-loop control structure as in an MCPS. For example, Da Silva et al. [14] propose an IDS that applies seven types of traffic-based rules to detect intruders: interval, retransmission, integrity, delay, repetition, radio transmission range and jamming. Ioannis et al. [17] propose a multitrust IDS with traffic-based collection that audits the forwarding behavior of suspects to detect blackhole and greyhole attacks launched by captured devices based on the the rate of specification violations.

Our contribution relative to prior work cited above is that we specifically consider behavior rules for MCPS actuators controlling patient treatment algorithms as well as for physiological sensors providing information concerning the physical environment. Further, we propose a methodology to transform behavior rules to a state machine, so that a device that is being monitored for its behavior can easily be checked against the transformed state machine for deviation from its behavior specification. Existing work [13], [15] only considered specification-based state machines for intrusion detection of communication protocol misbehaving patterns.

Untreated in the literature, in this paper we also investigate the impact of attacker behaviors on the effectiveness of MCPS intrusion detection. We demonstrate that our specification-based IDS technique can effectively trade higher false positives off for lower false negatives to cope with more sophisticated and hidden attackers. We show results for a range of configurations to illustrate this trade. Because the key motivation in MCPS is safety, our solution is deployed in a configuration yielding a high detection rate without compromising the false positive probability. Our approach is monitoring-based relying on the use of peer devices to monitor and measure the compliance degree of a trustee device connected to the monitoring node by the CPS network. The rules comparing monitor and trustee physiology (blood pressure, oxygen saturation, pulse, respiration and temperature) exceeds protection possible by considering devices in isolation.

Robert Mitchell and Ing-Ray Chen are with the Department of Computer Science, Virginia Polytechnic Institute and State University, Falls Church, VA, 22043.
E-mail: rrmitche@vt.edu and irchen@vt.edu.

The fundamental difference in designing IDSs for safety critical CPSs versus for other brands of systems is that the intrusion detection is closely tied with the physical components of the CPS, so the detection is less about communication protocol compliance but more about behavior compliance specific to the physical components to be controlled in the CPS. Thus, instead of monitoring packet routing or packet loss data for misbehavior detection of communication protocol compliance during packet transmission, IDSs for MCPSs may test medical sensor measurements and actuator settings for misbehavior detection of physical properties manifested because of attacks. For example, a patient requesting analgesic must have a pulse greater than some threshold, otherwise it may cause an overdose of analgesic delivered. Thus, if a patient requests analgesic while having a pulse below the threshold then an intruder may be involved. The behavior rules proposed in our work specifically address the expected behavior of individual physical components in the MCPS. The compliance threshold proposed in this paper specifically measures the goodness of a physical component. A challenge is to provide a high detection rate without introducing high false positives. We demonstrate that our IDS design based on the compliance threshold can effectively distinguish benign abnormalities from malicious attacks. To the best of our knowledge, there is no prior work discussing the difference between CPS intrusion detection and communication systems intrusion detection.

It is necessary to build an IDS per CPS domain/application since the behavior rules for specifying the behaviors of physical components/devices in a CPS are inherently domain/application specific.

In the literature, ISML [6] and T-Rex [29] are also specification-based approaches for intrusion detection in CPSs. However, none of them considered MCPSs. In the field of intrusion detection for MCPSs or healthcare systems, Asfaw et al. [4] studied an anomaly-based IDS for MCPSs. The authors focus on attacks that violate privacy of an MCPS; in contrast, our investigation focuses on attacks that violate the integrity of an MCPS. They use an anomaly-based approach while we use a specification-based approach. Asfaw et al. do not provide numerical results in the form of false negatives or positives which are the critical metrics for this research area; our investigation does provide these results.

Venkatasubramanian and Gupta [26] survey security solutions for pervasive healthcare applications. Like [4], the authors focus on attacks on a passive pervasive healthcare system that violate patient privacy while our investigation considers integrity attacks on an MCPS that harm a patient. Their countermeasures focus on encryption and authentication/access control.

Yang and Hwang [27] investigated an approach to fraud and abuse detection in healthcare applications. In contrast, our investigation focuses on the treatment, rather than the administrative, domain of healthcare. The authors use an anomaly-based approach while we use a specification-based approach. They provide numerical results that measure internal validity (the effectiveness of the data mining implementation) but do not provide externally valid metrics like *Receiver Operating Characteristic* (ROC) which can reveal the tradeoff between the detection rate vs. the false positive probability.

Porras and Neumann [22] study a hierarchical multitrust behavior-based IDS called Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) [13] using complementary signature based and anomaly-based analysis. The authors identify a signature-based analysis trade between the state space created/runtime burden imposed by rich rule sets and the increased false negatives that stem from a less expressive rule set. Porras and Neumann highlight two specific anomaly-based techniques using statistical analysis: one studies user sessions (to detect live intruders), and the other studies the runtime behavior of programs (to detect malicious code). EMERALD provides a generic analysis framework that is flexible enough to allow anomaly detectors to run with different scopes of multitrust data (service, domain or enterprise). However, Porras and Neumann did not report false positive or false negative probability data. While EMERALD pursues a domain-independent CPS security solution combining anomaly and signature-based analysis, our investigation focuses on one that is relevant for MCPSs using specification-based analysis.

Park et al. [21] propose a semi-supervised anomaly-based IDS targeted for assisted living environments. Their design is behavior-based and audits series of events which they call episodes. The authors' events are 3-tuples comprising sensor ID, start time and duration. Park et al. test data sets using four similarity functions based on: LCS, count of common events not in LCS, event start times and event durations They control episode length and similarity function as independent variables. The authors provide excellent ROC data which we use for a comparative analysis.

Tsang and Kwong [25] propose a multitrust IDS called Multi-agent System (MAS) that includes an analysis function called Ant Colony Clustering Model (ACCM). The authors intend for ACCM to reduce the characteristically high false positive rate of anomaly-based approaches while minimizing the training period by using an unsupervised approach to machine learning. MAS is hierarchical and contains a large number of roles: monitor agents collect audit data, decision agents perform analysis, action agents effect responses, coordination agents manage multitrust communication, user interface agents interact with human operators and registration agents manage agent appearance and disappearance. Their results indicate ACCM slightly outperforms the detection rates and significantly outperforms the false positive rates of k-means and expectation-maximization approaches. Like [22], MAS pursues a domain-independent CPS security solution using anomaly-based analysis; our investigation focuses on MCPS-specific IDS using specification-based analysis.

We will use Park et al. [21] and Tsang and Kwong [25] as base schemes against which BSID will be compared because no others provide meaningful $p_{\text{fp}}/p_{\text{fn}}$ data for a comparative analysis.

Our study of IDS warrants distinct treatment for medical versus generic CPSs because the behavior rule set we propose is application specific. CPSs in other domains will not have temperature sensors, medication dispensers or actuators supporting cardiac function. Furthermore, each CPS domain will have a unique environment: For example, while the population in an MCPS may be around 1000 based on the number of beds in a hospital, the population for a smart grid

CPS may be in the millions. Also, while the geography of a MCPS may span a single square kilometer based on the size of a medical campus, the area of operation for a unmanned air vehicle (UAV) may be thousands of $\mathrm{km}^2$.

## II. System Model

### A. Reference MCPS

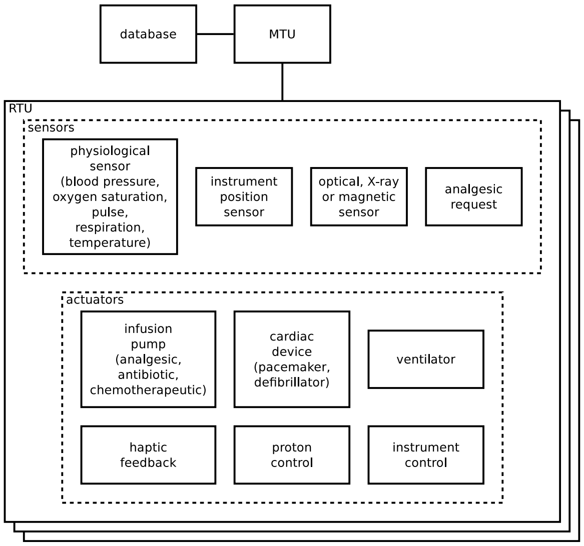
database
MTU
RTU
sensors
physiological sensor (blood pressure, oxygen saturation, pulse, respiration, temperature)
instrument position sensor
optical, X-ray or magnetic sensor
analgesic request
actuators
infusion pump (analgesic, antibiotic, chemotherapeutic)
cardiac device (pacemaker, defibrillator)
ventilator
haptic feedback
proton control
instrument control

Fig. 1. Medical Physical Components in the Reference MCPS.

We consider a pervasive health monitoring system embedding medical devices as our MCPS reference model as illustrated in Figure 1. For ease of disposition, we are particularly concerned with three types of sensor/actuator devices embedded in this MCPS: vital sign monitor (VSM), patient controlled analgesia (PCA) and cardiac device (CD). Many healthcare examples exist with these three devices. An example is an automated anesthesiologist where vital sign monitors (VSMs) sense patient physiology and control intravenous delivery of sedative. Specifically, VSMs sense respiration (Hz), oxygen saturation ($\mathrm{SpO_2}$), heart rate (Hz) and temperature (C). Another example is patient controlled analgesia (PCA) where patient analgesic requests and physiological sensor readings from VSMs drive infusion pumps [18]. A third example is an intensive care situation where the cardiac device (CD) frequency and pulse readings from VSMs drive biomedical devices such as a ventilator or automatic external defibrillator. The Welch Allyn Connex 6000 is the real system we base this investigation on. It has a 11.1 V 3.80 Ah (42 Wh) or 10.8 V 6.75 Ah (73 Wh) Li-ion battery, blood pressure sensor, thermometer, oxygen saturation sensor and two pulse rate sensors (one each integrated with blood pressure and oxygen saturation sensors). It uses USB for internal communication and IEEE 802.11 for external communication. It is reasonable to assume many medical devices use wireline communication, including PCA units. However, the VSM we consider in depth uses wireless communication. The CDs we consider in broad terms cannot use wireline communication, as they would permanently tether ambulatory, independent patients.

The IDS function is implemented in a distributed manner: every device is being monitored by other devices. There is no designated monitor node, so there is no single point of failure. For example, a VSM is being monitored by one or more peer VSMs which are themselves being monitored by other peer VSMs for security. (See Table I for monitor-trustee relationships.) The observations are collected by extracting audit data from logs generated by the relevant sensor or actuator drivers. This paper concerns peer-to-peer intrusion detection through behavior rule specification-based monitoring. The results obtained in this paper may be further used to implement a voting-based IDS to cope with node failure or colluding attacks [1], [9], [20].

### B. Threat Model

We focus on defeating inside attackers that violate the integrity of the MCPS with the objective to disable the MCPS functionality. Our design is also effective against attacks such as subtle manipulations that change medical doses slightly to cause long term harm to patients or medical or billing record exfiltrations which violate privacy. There are two distinct stages in an attack: before a node is compromised and after a node is compromised. Before a node is compromised, the adversary focuses on the tactical goal of achieving a foothold on the target system. Specifically, the adversary may use shellcode, code injection and capture attacks to compromise a physical component such as a VSM, PCA or CD. After a node is compromised, the now-inside attacker refocuses on the strategic goal of disabling the MCPS. Specifically, a compromised node may use data modification, forgery, greyhole/blackhole and replay attacks. In particular, a compromised sensor may return incorrect readings (thus performing modification attacks), and a compromised actuator may ignore control input (blackhole attacks), replay the previous command (replay attacks) or execute incorrect commands (forgery attacks).

We differentiate temporary system or environment abnormalities from malicious attacks by introducing a parameter, $p_{\mathrm{err}}$, to model the probability of a monitoring node misidentifying the state of a monitored node due to ambient noise. $p_{\mathrm{err}}$ varies significantly depending on the exact system being analyzed; it is an input parameter to reflect the level of ambient noise in the system. Based on the MCPS environment, the range of $p_{\mathrm{err}}$ can be measured a priori before the MCPS is put into operational use. $p_{\mathrm{err}}$ can vary based on location: It may be higher in outdoor MCPS environments in developed areas due to their significant cultural noise (e.g., terrestrial radio, mobile telephony and WiFi). However, multipath interference may cause it to be higher in indoor MCPS environments with RF-reflective construction. The abnormal states could result from $p_{\mathrm{err}}$ or from malicious attacks. We demonstrate that our IDS design based on compliance degree can effectively distinguish benign abnormalities from malicious attacks.

Real world attacks are emerging against MCPS components that cause node compromise. In particular, insulin pumps and cardiac devices are vulnerable [16]. MCPS attacks can occur through over the air software updates, stack buffer overflow exploits or logic bombs planted by third party software providers. If over the air software updates are frequent enough, an attacker can configure a radio with the appropriate frequency and demodulation technique, record updates, reverse engineer their format, craft a software load containing malware and deliver it to the target device. Source code analysis based on open-source software or disassembled and decompiled binaries can reveal stack buffer overflow vulnerabilities. If these methods are not available, the attacker can use fuzzing to prosecute this line of attack. Third party software logic bombs are a less viable form of MCPS attack. Due to their broad, unfocused distribution, the probability of detection and probability of attribution are relatively high. Medical devices, such as the VSM, PCA or CD we consider, detect these attacks by recording state information for the local node and peers, updating a state machine modeling the subject device and generating a detection when the automaton enters a malicious state. For example, if the heart rate component, as reported by the VSM, of the state machine indicates normal cardiac function, but the CD is in defibrillator mode, the IDS should generate a detection.

The security of wireless communication is handled by contemporary secret key technology such as PKI which provides authentication to prevent man-in-the-middle attacks, and the inherent unreliability of wireless communication is modeled by $p_{\text{err}}$ which accounts for ambient noise and unreliable wireless communication.

### C. Attacker Archetypes

We differentiate two attacker archetypes: reckless, random and opportunistic. A reckless attacker performs attacks whenever it has a chance to impair the MCPS functionality as soon as possible. A random attacker, on the other hand, performs attacks only randomly to avoid detection. It is thus insidious and hidden with the objective to cripple the MCPS functionality. We model the attacker behavior by a random attack probability $p_a$. When $p_a = 1$ the attacker is a reckless adversary. Random attacks are typically implemented with on-off attacks in real-world scenarios, so $p_a$ is not a random variable drawn from uniform distribution $U(0, 1)$ but rather a probability that a malicious node is performing attacks at any time with this on-off attack behavior. An opportunistic attacker is the third archetype. An opportunistic attacker exploits ambient noise modeled by $p_{\text{err}}$ (probability of mis-monitoring) to perform attacks. While a random attacker's $p_a$ is fixed, an opportunistic attacker decides its attack probability $p_a$ based on $p_{\text{err}}$ sensed. When $p_{\text{err}}$ is higher, the system is more vulnerable, so its $p_a$ is higher. An opportunistic attacker can be conservative or aggressive. We borrow from the demand-pricing relationship in the field of Economics [2], [8], [12], [28] to model the opportunistic attacker's attack probability $p_a$ as a function of $p_{\text{err}}$. Specifically, $p_a = \mathcal{C}p_{\text{err}}^{\varepsilon}$. With $C > 0$, this formula covers both conservative and aggressive attack behaviors: If $\varepsilon = 1$, $p_a$ increases linearly with $p_{\text{err}}$; this models a conservative opportunistic attacker; If $\varepsilon < 1$, $p_a$ increases exponentially with $p_{\text{err}}$; this models an aggressive opportunistic attacker, the extent of which is modeled by $\varepsilon$.

### D. Performance and Overhead Metrics

$p_{\text{fn}}$ is the false negative probability: the likelihood of misidentifying a bad node as good. Detection rate is the complement of false negative probability: $1 - p_{\text{fn}}$. $p_{\text{fp}}$ is the false positive probability: the likelihood of misidentifying a good node as bad. Receiver operating characteristic (ROC) graphs plot detection rate $(1 - p_{\text{fn}})$ as a function of false positive rate $(p_{\text{fp}})$.

The cost of overhead has three components: memory, processor load and communications channel usage. Memory is measured in terms of bytes, processor load is measured in terms of floating-point operations per second and communications channel usage is measured in terms of bits per second. The issue of overhead and complexity analysis of our IDS algorithm is not treated in this paper.

## III. MCPS Intrusion Detection Design

Security and functional modules are isolated from one another in our design. Vendors want to protect their intellectual property and maintain hard-earned certifications of their products; opening their designs and implementations threatens the former and allowing modification threatens the latter. The security community considers IDS isolation as the best practice in order to minimize the risk of compromise. Consequently, we envision a security module be added to a medical device but isolated from the medical device's functional modules.

### A. Behavior Rules

Behavior rules for a device are specified during the design and testing phase of an MCPS. Our intrusion detection protocol takes a set of behavior rules for a device as input and detects if a device's behavior deviates from the expected behavior specified by the set of behavior rules. Since the intrusion detection activity is performed in the background, it allows behavior rules to be changed if incomplete or imprecise specifications are discovered during the operational phase without disrupting the MCPS operation.

Our IDS design for the reference MCPS model relies on the use of lightweight specification-based *behavior rules* for each sensor or actuator medical device. They are oriented toward detecting an inside attacker attached to a specific physical component, provide a continuous (versus a binary) output between 0 and 1 (to account for transient faults and human errors) and allow a monitor device to perform intrusion detection on a neighboring trustee through monitoring. Here a monitor device is itself a sensor or monitor capable of doing intrusion detection on many trustees of different types. For example, a sensor might need to audit dissimilar sensors or even actuators for a small system. Therefore, a monitor device might have several sets of behavior rules (and thus several state machines), one for each trustee. Table I lists the MCPS behavior rules for PCA, CD and VSM. This table specifies the trustee and monitor devices for applying our IDS technique.

The behavior rule set specifies expected normal behaviors for each device and can detect deviation of normal behaviors regardless of the attacker's patterns. It does not rely on knowledge of known attacker patterns as in signature-based intrusion detection. However, behavior rules for a medical device will have to specify different acceptable parameter ranges to reflect the physiology and responses for different types of patients.

### *B. Transforming Rules to State Machines*

The following procedure transforms a behavior specification into a state machine: First, we identify the "attack state" as a result of a behavior rule being violated. Then, we transform this attack state into a conjunctive normal form predicate and identify the involved state components in the underlying state machine. Next, for each device, we combine the attack states into a Boolean expression in disjunctive normal form. Then we transform the union of all predicate variables into the state components of a state machine and establish their corresponding ranges. Finally we manage the number of states by state collapsing and identifying combinations of values that are not legitimate. Below we exemplify how a state machine is derived from the behavior specification in terms of behavior rules for the reference MCPS model.

Unsafe states in our state machine are not those "hazardous" states generated due to design faults (e.g., software bugs). Such "hazardous" states, once identified, would be removed as a result of design faults being identified and removed during the testing and debugging phase. The unsafe states (and safe states) in our approach are device-specific and are not removable because they are not caused by design faults. A CPS device will enter an unsafe state only when it is seen to deviate from the normal behavior specified by the behavior rule. This is the idea of our specification-based behavior rule intrusion detection. Here we note that while transitions into an unsafe state are not the direct result of system bugs, bugs and open doors are often the root cause that enables attackers to penetrate the system.

*1) Identify Attack States:* Attacks performed by a compromised sensor/actuator will drive the MCPS into certain attack states identifiable through analyzing the specification-based behavior rules.

For the PCA device, there are 4 attack states as a result of violating the 4 PCA behavior rules listed in Table I. No safety critical device is going to reach the market without safeguards in place. The behavior rules we propose for PCA do consider built-in safeguards; our rules add value by detecting anomaly behaviors not guarded by these built-in safeguards. The first PCA attack state is that a patient requesting analgesic has a pulse below some threshold. One way an attacker could exploit this is to cause an overdose of analgesic delivered by a PCA system. A patient will lose consciousness after receiving a sufficient amount of analgesic; if the PCA receives additional requests for analgesic, then an intruder is involved. The IDS can infer consciousness from pulse data. For this attack state, the PCA module is the trustee and the VSM is the monitor.

The second PCA attack state is that a patient requesting analgesic has a respiration rate below some threshold. A compromised PCA device performing this attack will drive the MCPS into this state. One way an attacker could exploit this is to cause an overdose of analgesic delivered by a PCA system. A patient will lose consciousness after receiving a sufficient amount of analgesic; if the PCA receives additional requests for analgesic, then an intruder is involved. The IDS can infer consciousness from respiration data. For this attack state, the PCA module is the trustee and the VSM is the monitor.

TABLE I
MCPS BEHAVIOR RULES

| Description | Trustee | Monitor |
|---|---|---|
| pulse above threshold during analgesic request | PCA | VSM |
| respiration above threshold during analgesic request | PCA | VSM |
| analgesic request rate below safe threshold | PCA | VSM |
| no analgesic infusion during defibrillation | PCA | VSM |
| pulse matches pacemaker frequency | CD | VSM |
| patient is unstable before defibrillation | CD | VSM |
| trustee blood pressure matches monitor | VSM | peer VSM |
| trustee oxygen saturation matches monitor | VSM | peer VSM |
| trustee pulse matches monitor | VSM | peer VSM |
| trustee respiration matches monitor | VSM | peer VSM |
| trustee temperature matches monitor | VSM | peer VSM |

The third PCA attack state is that an analgesic request rate exceeds some threshold. One way an attacker could exploit this is to cause an overdose of analgesic delivered by a PCA system. It is important to distinguish physical button presses from requests actually generated. While a patient in pain may press the button more frequently than is safe due to pain, the PCA module should only fulfill requests within the safe threshold. If the PCA module fulfills requests too frequently, then an intruder is involved. For this attack state, the PCA module is the trustee and the VSM is the monitor.

The fourth PCA attack state is that the PCA infusion rate, $x$, is in (0, 100%] and the cardiac device mode, $y$, is defibrillation, yielding a state with two components. As the device being evaluated transitions from one state $(x_0, y_0)$ to another $(x_1, y_1)$, the monitor can check if $(x_0, y_0)$ and $(x_1, y_1)$ are both good states. For this attack state, the PCA module is the trustee and the VSM is the monitor.

For the CD device, there are two attack states. The first CD attack state is that pulse average is not equal to CD frequency when acting as a pacemaker. One way an attacker could exploit this is to change the pacemaker frequency. If the CD frequency when acting as a pacemaker is substantially different from the patient's heart rate, then an intruder is involved. The trustee in this case is the CD. For this attack state, the CD is the trustee and the VSM is the monitor.

The second CD attack state is that pulse average is within a normal range when the CD enters defibrillator mode. One way an attacker could exploit this is to defibrillate a stable patient. If the CD enters defibrillator mode unnecessarily, then an intruder is involved. For this attack state, the CD is the trustee and the VSM is the monitor.

For the VSM device, there are 5 attack states in which a trustee sensor reading (blood pressure, oxygen saturation, pulse, respiration, or temperature) is beyond 100% of the corresponding monitor sensor reading. A peer VSM in the neighborhood of the trustee sensor serves as the monitor, measuring the same physical phenomenon. In this rule there is a variable "sensor reading % deviation" which can go from 0 to 100% in 10% increments, yielding 11 possible values. The monitor observing a trustee sensor will check the status of this variable. As the trustee sensor goes from one state to another, say, from 10 to 20%, the monitor will assess the deviation of good behaviors of the trustee by means of host IDS techniques.

*2) Express Attack States in Conjunctive Normal Form:* Table II lists the attack states in Conjunctive Normal Form.

*3) Consolidate Predicates in Disjunctive Normal Form:*

*a) PCA:* ((Analgesic Request = TRUE) $\wedge$ (Pulse $< T$)) $\vee$ ((Analgesic Request = TRUE) $\wedge$ (Respiration $< T$)) $\vee$ (Analgesic Request Rate $> T$) $\vee$ ((Analgesic Infusion Rate

TABLE II
ATTACK STATES IN CONJUNCTIVE NORMAL FORM

| |
|---|
| (Analgesic Request = TRUE) $\wedge$ (Pulse $< T$) |
| (Analgesic Request = TRUE) $\wedge$ (Respiration $< T$) |
| Analgesic Request Rate $> T$ |
| (Analgesic Infusion Rate $> 0$) $\wedge$ (Mode = DEFIBRILLATOR) |
| (Mode = PACEMAKER) $\wedge$ ($\vert$Pulse - Pacemaker Frequency$\vert > \delta$) |
| (Mode = DEFIBRILLATOR) $\wedge$ ($L <$ Pulse $< H$) |
| $\vert$Monitor Blood Pressure - Trustee Blood Pressure$\vert > \delta$ |
| $\vert$Monitor Oxygen Saturation - Trustee Oxygen Saturation$\vert > \delta$ |
| $\vert$Monitor Pulse - Trustee Pulse$\vert > \delta$ |
| $\vert$Monitor Respiration - Trustee Respiration$\vert > \delta$ |
| $\vert$Monitor Temperature - Trustee Temperature$\vert > \delta$ |

TABLE III
MCPS STATE COMPONENTS

| Name | Control or Reading | Range |
|---|---|---|
| Analgesic Request | Reading | true, false |
| Pulse | Reading | [0, 240 bpm] |
| Respiration | Reading | [0, 60 bpm] |
| Analgesic Request Rate | Reading | [0, 4/hour] |
| Blood Pressure | Reading | [0, 240 mmHg]<br>× [0, 160 mmHg] |
| Oxygen Saturation | Reading | [0, 100%] |
| Temperature | Reading | [32, 42 C] |
| Analgesic Infusion Rate | Control | [0, 100%] |
| Mode | Control | passive, pacemaker, defibrillator |
| Pacemaker Frequency | Control | [0, 240 bpm] |

$> 0$) $\wedge$ (Mode = DEFIBRILLATOR))

*b) CD:* ((Analgesic Infusion Rate $> 0$) $\wedge$ (Mode = DEFIBRILLATOR)) $\vee$ ((Mode = PACEMAKER) $\wedge$ ($\vert$Pulse - Pacemaker Frequency$\vert > \delta$)) $\vee$ ((Mode = DEFIBRILLATOR) $\wedge$ ($L <$ Pulse $< H$))

*c) VSM:* ($\vert$Monitor Blood Pressure - Trustee Blood Pressure$\vert > \delta$) $\vee$ ($\vert$Monitor Oxygen Saturation - Trustee Oxygen Saturation$\vert > \delta$) $\vee$ ($\vert$Monitor Pulse - Trustee Pulse$\vert > \delta$) $\vee$ ($\vert$Monitor Respiration - Trustee Respiration$\vert > \delta$) $\vee$ ($\vert$Monitor Temperature - Trustee Temperature$\vert > \delta$)

*4) Identify State Components and Component Ranges:* We quantize continuous components at integer scale in permissible ranges. For example, pulse is in the range of [0, 240 bpm] and respiration is in the range of [0, 60 bpm]. Table III shows a complete list of the permissible ranges of MCPS state components. The resulting PCA automaton has $2 \times 241 \times 61 \times 5 \times 101 \times 3 = 4.454 \times 10^7$ states. The resulting CD automaton has $241 \times 101 \times 3 \times 241 = 1.760 \times 10^7$ states. The resulting VSM automaton has $241 \times 161 \times 241 \times 161 \times 101 \times 101 \times 241 \times 241 \times 61 \times 61 \times 11 \times 11 = 4.016 \times 10^{23}$ states. All of these automata are too large; we deal with this state explosion in the next step.

*5) Manage State Space:* To manage the number of states, we reduce the size of the state machine by abbreviating the values for some components. For the PCA device, only three values are relevant for pulse, respiration and analgesic request rate: normal, beyond warning threshold and beyond unsafe threshold. Therefore, we collapse the domain for each of these components to three values. Likewise only two values are relevant for analgesic infusion rate: zero or nonzero. Therefore, we collapse the domain for this component to two values. This treatment yields a modest PCA state machine with $2 \times 3 \times 3 \times 3 \times 2 \times 3 = 324$ states. 50 of these states are safe because they fully comply with all of the behavior rules from Table I. 80 are warning states because they exceed the warning threshold for at least one behavior rule. 194 of these states are unsafe because they violate or exceed the unsafe threshold for at least one of the behavior rules. Rather than their values, the VSM behavior rules only need to know whether each vital sign trustee reading matches, is farther than the warning threshold or is farther than the unsafe threshold from the corresponding monitor reading. Therefore, we collapse the domain for each of these components to three values. This treatment yields a modest VSM state machine with $3 \times 3 \times 3 \times 3 \times 3 = 243$ states. One of these states is safe because the monitor and trustee readings match for all five components as described in Table I. 31 are warning states because the monitor and trustee readings differ by more than the warning margin for at least one component but not more than the unsafe threshold for any component. 211 of these states are unsafe because at least one component differs by more than the unsafe threshold.

*6) Behavior Rule State Machines:* Here we describe how to generate the behavior rule state machine of a medical device. We use the VSM device as an example. The VSM state machine consisting of one safe, 31 warning and 211 unsafe states based on the behavior rules is generated as follows. First we label these states as $1, 2, \ldots, n = 243$. Next we assign $p_{ij}$, the probability that state $i$ goes to state $j$, for each $(i, j)$ pair in the state machine to reflect a good or bad VSM's behavior.

A good VSM should stay in safe states 100% of the time. This will give the compliance degree of a good VSM close to one. However, occasionally it may be detected by the monitor node as staying in a warning or unsafe state due to ambient noise resulting from unexpected environment or system condition changes, as well as wireless communication faults. Let $p_{\mathrm{err}}$ be the error probability of a monitor node misidentifying the status of a trustee node due to ambient noise and wireless communication faults. During the testing phase, we seed a good VSM in the system and assign a monitor node to observe and measure $p_{ij}$ of the good VSM in the presence of the error probability $p_{\mathrm{err}}$: $p_{ij}$ is $p_{\mathrm{err}} \times 31/(31+211)$ when $j$ is one of the 31 warning states, $p_{ij}$ is $p_{\mathrm{err}} \times 211/(31+211)$ when $j$ is one of the 211 unsafe states, and $p_{ij}$ is $1 - p_{\mathrm{err}}$ when $j$ is the one good state. Figure 2 illustrates the behavior rule state machine for a good VSM in the MCPS. One dotted slash and crossed dotted slashes over a state indicate a warning state and an unsafe state, respectively. Transitions into states covered with a dotted slash are valid, but their marginality is cause for concern. Transitions into states covered with crossed dotted slashes are invalid and cause for an alert. All transitions are possible. Each state component represents how one of the trustee node attributes matches its counterpart from the monitor. For the VSM device, pulse, blood pressure, oxygen saturation, temperature and respiration are the device attributes of interest. Note that each device has its own state machine with device-specific attributes being the state components of the state machine.

For a compromised VSM, $p_{ij}$ depends on its attacker type: A reckless attacker presumably will stay in unsafe or warning states 100% of the time; however, occasionally it may be detected by the monitor node as staying in a safe state due to ambient noise and wireless communication faults. During the testing phase, we seed a reckless attacker in the system

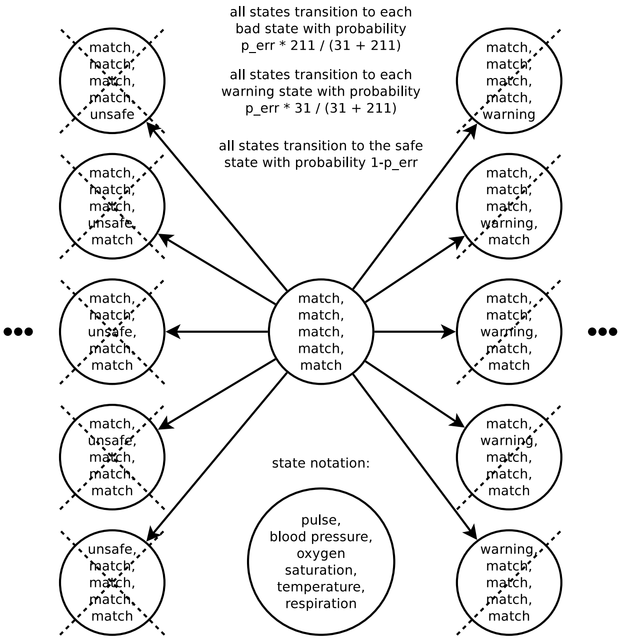
all states transition to each
bad state with probability
p_err * 211 / (31 + 211)
all states transition to each
warning state with probability
p_err * 31 / (31 + 211)
all states transition to the safe
state with probability 1-p_err
match, match, match, match, unsafe
match, match, match, unsafe, match
match, match, unsafe, match, match
match, unsafe, match, match, match
unsafe, match, match, match, match
match, match, match, match, match
match, match, match, match, warning
match, match, match, warning, match
match, match, warning, match, match
match, warning, match, match, match
warning, match, match, match, match
•••
•••
state notation:
pulse, blood pressure, oxygen saturation, temperature, respiration

Fig. 2. Good VSM Behavior Rule State Machine.

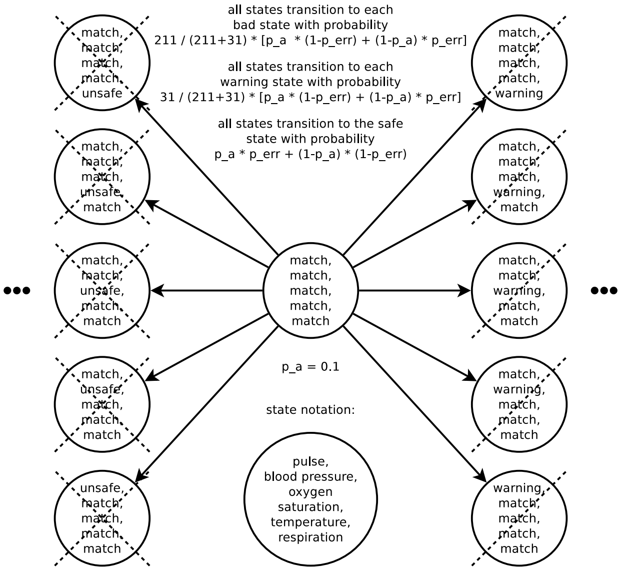
all states transition to each
bad state with probability
211 / (211+31) * [p_a * (1-p_err) + (1-p_a) * p_err]
all states transition to each
warning state with probability
31 / (211+31) * [p_a * (1-p_err) + (1-p_a) * p_err]
all states transition to the safe
state with probability
p_a * p_err + (1-p_a) * (1-p_err)
match, match, match, match, unsafe
match, match, match, unsafe, match
match, match, unsafe, match, match
match, unsafe, match, match, match
unsafe, match, match, match, match
match, match, match, match, match
match, match, match, match, warning
match, match, match, warning, match
match, match, warning, match, match
match, warning, match, match, match
warning, match, match, match, match
•••
•••
p_a = 0.1
state notation:
pulse, blood pressure, oxygen saturation, temperature, respiration

Fig. 3. Random Attacker VSM Behavior Rule State Machine.

following its attacker profile and assign a monitor node to observe and measure $p_{ij}$: $p_{ij}$ is $211/(211+31) \times (1 - p_{\text{err}})$ when $j$ is one of the 211 unsafe states, $31/(211+31) \times (1 - p_{\text{err}})$ when $j$ is one of the 31 warning states, and $p_{\text{err}}$ when $j$ is the one good state where $p_{\text{err}}$ of the error probability of misidentifying the status of a reckless attacker due to ambient noise and wireless communication faults. For a random attacker with attack probability $p_a$, $p_{ij}$ is $211/(211+31) \times (p_a \times (1-p_{\text{err}}) + (1-p_a) \times p_{\text{err}})$ when $j$ is one of the 211 bad states, $31/(211+31) \times (p_a \times (1-p_{\text{err}}) + (1-p_a) \times p_{\text{err}})$ when $j$ is one of the 31 warning states, and $p_a \times p_{\text{err}} + (1-p_a) \times (1-p_{\text{err}})$ when $j$ is the one good state. We note that a random attacker with attack probability $p_a$ will stop attacking with probability $1 - p_a$, which will be detected by the monitor node with probability $1 - p_{\text{err}}$. Figure 3 illustrates the behavior rule state machine for a random attacker VSM in the MCPS.

### C. Collect Compliance Degree Data

Compliance degree is the extent to which a device behaves securely, and we propose a number of ways to measure this property in this section. Compliance degree data is a time series of compliance degree measurements for a device; this is the input stream that triggers detections. We use the state machines to collect compliance degree data of a good and a bad medical device during the system testing and debugging phase before deployment. The behaviors of a good and a bad device performing random attacks are simulated and compliance degree data are collected to allow us to predict the false positive and false negative probabilities. While we experimented with a range of configurations, our solution is deployed with settings yielding a high detection rate because the key motivation in MCPS is safety.

Specifically, we profile the analgesic request, pulse, respiration, blood pressure, oxygen saturation, temperature, analgesic infusion rate, cardiac device mode and pacemaker frequency, given that they are being controlled by a good or a bad medical device.

We model the behavior of a medical device by a stochastic process such that it may be in state 0, 1, 2, . . . , $n$ in a state machine for intrusion detection of this medical device, with $p_{ij}$ parameterized as discussed earlier in Section III-B6. Then, the probability that the stochastic process is in state $j$ is given by:

$$\pi_j = \sum_{i=0}^{n} \pi_i p_{ij} \tag{1}$$

Since we have $n+1$ states, we have $n+1$ equations above, one for each state. This will yield infinite solutions, so we replace one equation with:

$$\sum_{i=0}^{n} \pi_i = 1 \tag{2}$$

The physical meaning of $\pi_j$ is the probability that a device is in state $j$ at any time.

Let $c$ be the compliance degree of a node. With the above formulation, it is calculated as the sum of the products of the each state's grade and probability:

$$c = \sum_j c^j \pi_j \tag{3}$$

where $c^j$ is the "grade" assignment to state $j$, measuring the closeness between the observed behavior (in state $j$) and the specified "good" behavior. We consider two grading strategies: binary and distance-based. For binary grading, we assign a value of 1 to state $j$ if it is secure and a value of 0 otherwise:

$$c^j = \begin{cases} 1 & \text{if state } j \text{ is a secure state} \\ 0 & \text{otherwise} \end{cases}$$

With binary grading, the compliance degree $c$ of a device essentially is equal to the proportion of the time the device is in secure states.

For distance-based grading, we still assign a value of 1 to state $j$ if it is secure. However, if state $j$ is insecure, we assign it a value in $[0, 1]$ representing the distance of state $j$ from a secure state. Therefore, $c^j$ is assigned as follows:

$$c^j = 1 - \text{distance}_j/\text{maximum distance}$$

where distance$_j$ is the distance between state $j$ and the nearest secure state, and maximum distance is the longest distance between any insecure state and the nearest secure state in the state machine. By this assignment, if state $j$ is secure, distance$_j$ is zero, hence $c^j = 1$. If state $j$ is insecure, $c^j$ is still close to 1 if $j$ is close to a secure state but is close to 0 if $j$ is far from a secure state. With $c^j$ assigned, we can then calculate the compliance degree, $c$, of a node in state machine $s$ using Equation 3 where $\pi_j$ gives the proportion of time a node stays in $j$ over the observation period. We study six distance-based grading strategies for measuring $c^j$ and for computing the compliance degree based on Equation 3.

- Hamming distance, also called signal distance, applies to a pair of multidimensional data points. This is the number of state components that differ between two sequences: in our application, state $j$ and the closest secure state. $c^j = 1 - \text{Hamming}(j)/\max(\text{Hamming}(\cdot))$. For example, consider a system with one state component with two possible values. This system has two states: $0, 1$. Consider state 0 safe and state 1 unsafe. Therefore, $c^0 = 1 - 0/1 = 1$ and $c^1 = 1 - 1/1 = 0$. Consider a node with $\pi_0 = 0.9$ and $\pi_1 = 0.1$. Therefore, for this node, $c = 1 \times 0.9 + 0 \times 0.1 = 0.9$.
- Manhattan distance, also called rectilinear distance, applies to a pair of multidimensional data points. This is the sum of the differences between state components of two sequences: in our application, state $j$ and the closest secure state. $c^j = 1 - \text{Manhattan}(j)/\max(\text{Manhattan}(\cdot))$. For example, consider a system with two state components with two possible values. This system has four states: $00, 01, 10, 11$. Consider state $00$ safe and the rest unsafe. Therefore, $c^{00} = 1 - 0/2 = 1$, $c^{01} = 1 - 1/2 = 0.5$, $c^{10} = 1 - 1/2 = 0.5$ and $c^{11} = 1 - 2/2 = 0$. Consider a node with $\pi_{00} = 0.9$, $\pi_{01} = 0.045$, $\pi_{10} = 0.045$ and $\pi_{11} = 0.01$. Therefore, for this node, $c = 1 \times 0.9 + 0.5 \times 0.045 + 0.5 \times 0.045 + 0 \times 0.01 = 0.945$.
- Euclidean distance applies to a pair of multidimensional data points. This is the square root of the sum of the squares of the state component differences between two sequences: in our application, state $j$ and the closest secure state. $c^j = 1 - \text{Euclidean}(j)/\max(\text{Euclidean}(\cdot))$. For example, consider a system with two state components with two possible values. This system has four states: $00, 01, 10, 11$. Consider state $00$ safe and the rest unsafe. Therefore, $c^{00} = 1 - \sqrt{0^2 + 0^2}/\sqrt{2} = 1$, $c^{01} = 1 - \sqrt{0^2 + 1^2}/\sqrt{2} = 0.707$, $c^{10} = 1 - \sqrt{1^2 + 0^2}/\sqrt{2} = 0.707$ and $c^{11} = 1 - \sqrt{1^2 + 1^2}/\sqrt{2} = 0$. Consider a node with $\pi_{00} = 0.9$, $\pi_{01} = 0.045$, $\pi_{10} = 0.045$ and $\pi_{11} = 0.01$. Therefore, for this node, $c = 1 \times 0.9 + 0.707 \times 0.045 + 0.707 \times 0.045 + 0 \times 0.01 = 0.964$.
- Longest common subsequence (LCS) distance, not to be confused with longest common substring distance, applies to a pair of time series. Longest common subsequence differs from longest common substring because a common subsequence does not need to be contiguous; extra values can appear within a common subsequence. $c = \text{LCS}(\text{monitor}, \text{trustee})/\text{time series length}$. For example, consider a system using a two point time series with two possible values. Monitor and trustee time series have four possible values: $00, 01, 10, 11$. If we consider a monitor time series of $00$ and a trustee time series of $00$, $c = 2/2 = 1$. If we consider a monitor time series of $00$ and a trustee time series of $10$, $c = 1/2 = 0.5$. If we consider a monitor time series of $00$ and a trustee time series of $11$, $c = 0/2 = 0$.
- Levenshtein distance, also called edit distance, applies to a pair of time series. This is the minimum number of edits required to transform one sequence into another; Levenshtein edits comprise insertion, deletion and substitution. $c = 1 - \text{Levenshtein}(\text{monitor}, \text{trustee})/\text{time series length}$. For example, consider a system using a three point time series with three possible values. Monitor and trustee time series have 27 possible values: $000, 001, 002, \ldots, 222$. If we consider a monitor time series of $000$ and a trustee time series of $000$, $c = 1 - 0/3 = 1$. If we consider a monitor time series of $012$ and a trustee time series of $120$, $c = 1 - 2/3 = 0.333$. If we consider a monitor time series of $000$ and a trustee time series of $111$, $c = 1 - 3/3 = 0$.
- Damerau-Levenshtein is based on Levenshtein but adds transposition of two contiguous values to the set of allowed edits. For example, consider a system using a three point time series with three possible values. Monitor and trustee time series have 27 possible values: $000, 001, 002, \ldots, 222$. If we consider a monitor time series of $000$ and a trustee time series of $000$, $c = 1 - 0/3 = 1$. If we consider a monitor time series of $012$ and a trustee time series of $021$, $c = 1 - 1/3 = 0.667$. If we consider a monitor time series of $000$ and a trustee time series of $111$, $c = 1 - 3/3 = 0$.

A system manager can select the best grading strategy based on performance results depending on the application environment. In Section VI, we provide an example to illustrate the utility.

### D. Compliance Degree Distribution

The measurement of compliance degree of a device is not perfect and can be affected by noise and unreliable wireless communication in the MCPS. We model the compliance degree by a random variable $X$ with $G(\cdot) = Beta(\alpha, \beta)$ distribution [23]. In probability theory and statistics, the $Beta$ distribution is a family of continuous probability distributions defined on the interval [0, 1], suitable for modeling the random behavior of percentages and proportions. The value 0 indicates that the output is totally unacceptable (zero compliance) and 1 indicates the output is totally acceptable (perfect compliance), such that $G(a)$, $0 \le a \le 1$, is given by

$$G(a) = \int_0^a \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}\, dx \tag{4}$$

and the expected value of $X$ is given by

$$E_B[X] = \int_0^1 x \, \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}\, dx = \frac{\alpha}{\alpha+\beta} \tag{5}$$

The $\alpha$ and $\beta$ parameters are to be estimated based on the method of maximum likelihood by using the compliance degree history collected $(c_1, c_2, \ldots, c_n)$ during the system's testing phase. The maximum likelihood estimates of $\alpha$ and $\beta$ are obtained by numerically solving the following equations:

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

$$\frac{n\frac{\partial\Gamma(\hat{\alpha}+\hat{\beta})}{\partial\hat{\beta}}}{\Gamma(\hat{\alpha}+\hat{\beta})} - \frac{n\frac{\partial\Gamma(\hat{\beta})}{\partial\hat{\beta}}}{\Gamma(\hat{\alpha})} + \sum_{i=1}^{n}\log(1-c_i) = 0 \qquad (6)$$

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 $\alpha$ 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 $\beta$ is

$$\hat{\beta} = \frac{n}{\sum_{i=1}^{n}\log(\frac{1}{1-c_i})} \qquad (7)$$

The reason we choose the $Beta$ distribution as described above is that the domain of the $Beta$ distribution can be viewed as a probability, so it can be used to describe the prior distribution over the probability (of a distribution) which models the node compliance degree. By applying Bayesian inference, the $Beta$ distribution then can be used as the posterior distribution of the probability after observing sufficient instances.

### *E. False Negative and Positive Probabilities*

Our intrusion detection is characterized by false negative and positive probabilities, denoted by $p_{\text{fn}}$ and $p_{\text{fp}}$, respectively. A false negative occurs when a bad medical device is missed as good, while a false positive occurs when a good medical device is misdiagnosed as bad. While neither is desirable, a false negative in an MCPS is especially impactful to the patient's well being. Because the key motivation in MCPS is safety, we searched for a configuration yielding a high detection rate without compromising the false positive probability. In this paper we consider a threshold criterion. That is, if a bad node's compliance degree denoted by $X_b$ with a probability distribution obtained by Equation 4 above is higher than a system minimum compliance threshold $C_T$ then there is a false negative. Suppose that the compliance degree $X_b$ of a bad node is modeled by a $G(\cdot) = Beta(\alpha, \beta)$ distribution as described above. Then the host IDS false negative probability $p_{\text{fn}}$ is given by:

$$p_{\text{fn}} = \Pr\{X_b > C_T\} \ = 1 - G(C_T). \qquad (8)$$

On the other hand, if a good node's compliance degree denoted by $X_g$ is less than $C_T$ then there is a false positive. Again suppose that the compliance degree $X_g$ of a good node is modeled by a $G(\cdot) = Beta(\alpha, \beta)$ distribution. Then the host false positive probability $p_{\text{fp}}$ is given by:

$$p_{\text{fp}} = \Pr\{X_g \leq C_T\} = G(C_T). \qquad (9)$$

## IV. Simulation

We collect compliance degree history $c_1, c_2, \ldots, c_n$ of a device by means of Monte Carlo simulation. Monte Carlo simulation allows us to generate repeated random sampling following the stochastic process of a devices state machine to obtain numerical results. We use the VSM device in the reference MCPS defined in Section II to exemplify the utility of our IDS technique for secure MCPS applications. The Welch Allyn Connex 6000 is an example of a VSM that fits into our model.

Specifically we simulate the procedure described in Section III-B6 to construct the state machines of a good VSM device and a bad VSM device. For a good VSM device, we simulate $p_{ij}$ (see III-B6 for its definition) as $1 - p_{\text{err}}$ when $j$ is the single good state, and as $p_{\text{err}}$ when $j$ is one of the 242 bad states (treating both 31 warning and 211 unsafe states as bad). For a bad VSM device with random attack probability $p_a$, we simulate $p_{ij}$ as $((1-p_a)\times(1-p_{\text{err}})+p_a\times p_{\text{err}})$ when $j$ is the single good state, and as $(p_a\times(1-p_{\text{err}})+(1-p_a)\times p_{\text{err}})/242$ when $j$ is one of the 242 bad states.

Given the state machine of a VSM device generated above, we collect a sequence of compliance degree values $(c_1, c_2, \ldots, c_n)$ with $n = 1000$ Monte Carlo simulation test runs. In each simulation test run, we start from state 0 and then follow the stochastic process of this device as it goes from one state to another. We continue doing this until at least one state is traversed sufficiently (say 100 times). Then we calculate the limiting probability that the device is in state $j$, $\pi_j$, using the ratio of the number of transitions leading to state $j$ to the total number of state transitions. Then we collect one instance of $c$ using Equation 3. We repeat a sufficiently large $n = 1000$ test runs to collect $c_1, c_2, \ldots, c_n$ needed for computing the distribution of the compliance degree of a good or a bad medical device performing reckless or random attacks.

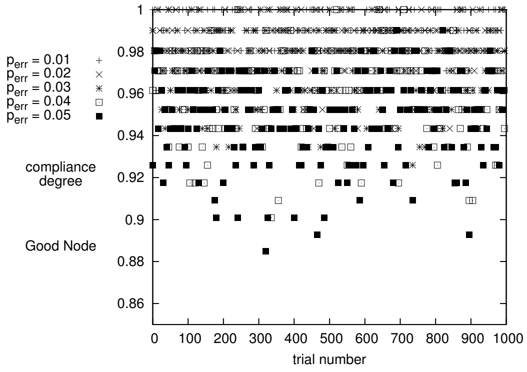

Fig. 4. Sensitivity of Good Node Compliance Degree to $p_{\text{err}}$.

Figure 4 shows compliance degree raw data with $X = 1, 2, \ldots, n$ and $Y = c_1, c_2, \ldots, c_n$, for $n = 1000$ points, for a good VSM node with several $p_{\text{err}}$ values. There are five clouds of compliance degree data, one corresponding with each $p_{\text{err}}$ setting. We see that as $p_{\text{err}}$ (representing ambient noise) increases, the cloud of compliance degree data moves down, i.e., the compliance degree of the good node decreases. This is because as the noise increases, there is a higher probability of the monitoring node misidentifying the good state status of the good VSM node.

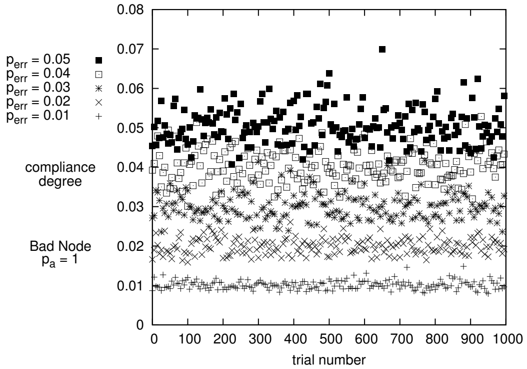

Fig. 5. Sensitivity of Bad Node Compliance Degree to $p_{\rm err}$ for Reckless Attackers.

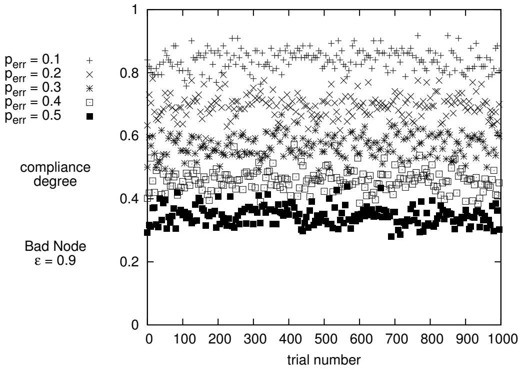

Fig. 6. Sensitivity of Bad Node Compliance Degree to $p_{\rm err}$ for Opportunistic Attackers.

Figure 5 shows the sensitivity of $c_1, c_2, \ldots, c_n$ to $p_{\rm err}$ for a bad VSM node attacking recklessly. Like Figure 4, there are five clouds of compliance degree data, one corresponding with each $p_{\rm err}$ setting. However, in this case as $p_{\rm err}$ increases, the cloud of compliance degree data moves up, i.e., the compliance degree of the bad VSM node increases. This is because as the noise increases, there is a higher probability of the monitoring node misidentifying the bad state status of the bad VSM node.

Figure 6 shows the sensitivity of $c_1, c_2, \ldots, c_n$ to $p_{\rm err}$ for a bad VSM node attacking opportunistically (with $\varepsilon = 0.9$). Like Figure 5, there are five clouds of compliance degree data, one corresponding with each $p_{\rm err}$ setting where higher $p_{\rm err}$ correlates with lower compliance. We see the compliance of of the opportunistic attacker is more sensitive to $p_{\rm err}$ than the reckless attacker: While the range of compliance covers (0.3, 0.9) for the opportunistic attacker, it is limited to approximately (0.01, 0.07) for the reckless attacker. Also, while the variance in compliance remains constant for the opportunistic attacker, the same quantity increases with $p_{\rm err}$ for the reckless attacker.

Figure 7 shows the sensitivity of $c_1, c_2, \ldots, c_n$ to $p_{\rm a}$, the random attack probability by a bad node. There are five clouds of compliance degree data, one corresponding with each $p_a$ setting. As $p_a$ increases, the cloud of compliance degree data moves down, i.e., the bad node's compliance degree decreases. This is because as the bad VSM node performs more frequent attacks, it is more easily to be detected, so its measured compliance degree decreases.

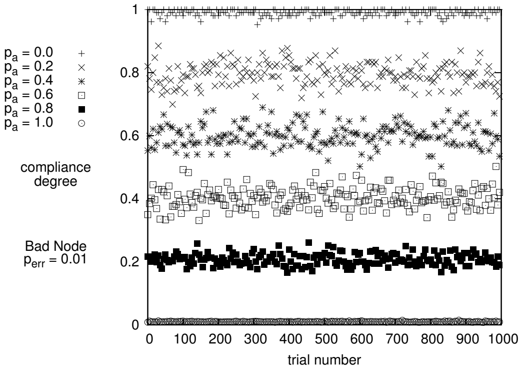

Fig. 7. Sensitivity of Bad Node Compliance Degree to $p_a$ for Random Attackers.

TABLE IV
$\beta$ IN BETA(1,$\beta$) AND RESULTING $p_{\rm fn}$ AND $p_{\rm fp}$ VALUES UNDER VARIOUS RANDOM ATTACK MODELS FOR VSM ($C_T = 0.9, p_{\rm err} = 0.01$).

| Attack Type | $\beta$ | $p_{\rm fn}$ | $p_{\rm fp}$ |
|---|---|---|---|
| Reckless Attacker ($p_a = 1.00$) | 98.5 | 0.001% | 14.8% |
| Random Attacker ($p_a = 0.80$) | 4.29 | 0.005% | 14.8% |
| Random Attacker ($p_a = 0.40$) | 1.08 | 8.33% | 14.8% |
| Random Attacker ($p_a = 0.20$) | 0.621 | 23.9% | 14.8% |
| Random Attacker ($p_a = 0.10$) | 0.441 | 36.3% | 14.8% |

With $c_1, c_2, \ldots, c_n$ of a good or bad VSM device in hand, we apply Equation 7 to compute the $\beta$ parameter value of $G(\cdot) = Beta(\alpha, \beta)$ for the probability distribution of the compliance degree for a good or a bad VSM device. We then calculate $p_{\rm fn}$ and $p_{\rm fp}$ by Equations 8 and 9, respectively, given the minimum compliance degree $C_T$ as input reflecting the consequence of false negatives over false positives for the VSM device. For an MCPS we prioritize achieving a low false negative probability because the key motivation is safety.

Table IV shows the $\beta$ values and the resulting $p_{\rm fn}$ and $p_{\rm fp}$ values when $C_T = 0.9$, $p_{\rm err} = 0.01$, and the binary grading strategy is being used to assign $c^j$ to state $j$ for a reckless or random attacker. $C_T$ is a design parameter to be fine-tuned to trade high false positives for low false negatives due to safety criticality as described below. We observe that when the random attack probability $p_a$ is high, the attacker can be easily detected, as evidenced by a low false negative probability. Especially when $p_a = 1$, a reckless attacker can hardly be missed. On the other hand, as $p_a$ decreases, the attacker becomes more hidden and insidious and the false negative probability increases. The false positive probability remains the same regardless of the random attack probability because it is a metric measuring the detection error against a good node only.

Likewise, Table V shows the $\beta$ values and the resulting

TABLE V
$\beta$ IN BETA(1,$\beta$) AND RESULTING $p_{\rm fn}$ AND $p_{\rm fp}$ VALUES UNDER VARIOUS OPPORTUNISTIC ATTACK MODELS FOR VSM ($C_T = 0.9, p_{\rm err} = 0.01, \mathcal{C} = 10$).

| Attack Type | $\beta$ | $p_{\rm fn}$ | $p_{\rm fp}$ |
|---|---|---|---|
| Aggressive Attacker with $\varepsilon = 0.8$ | 0.723 | 18.9% | 14.8% |
| Aggressive Attacker with $\varepsilon = 0.9$ | 0.545 | 28.5% | 14.8% |
| Conservative Attacker with $\varepsilon = 1.0$ | 0.441 | 36.3% | 14.8% |

$p_{\text{fn}}$ and $p_{\text{fp}}$ values when $C_T = 0.9$, $p_{\text{err}} = 0.01$, and the binary grading strategy is being used to assign $c^j$ to state $j$ for an opportunistic attacker. We observe that as $\varepsilon$ decreases, the opportunistic attacker can be detected more easily because of its more aggressive attack behavior.

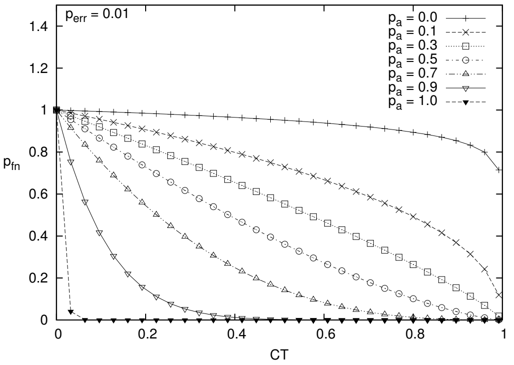

Fig. 8. Probability of False Negative vs. Compliance Threshold for Detecting Random Attackers.

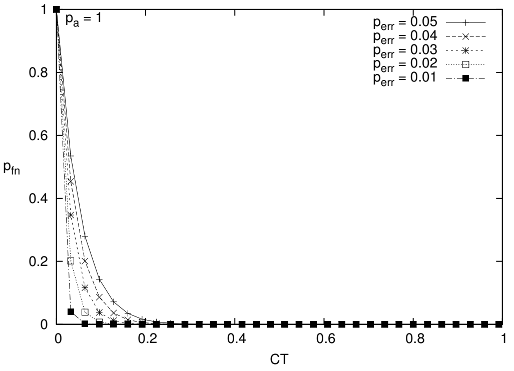

Fig. 9. Probability of False Negative vs. Compliance Threshold and $p_{\text{err}}$ for Detecting Reckless Attackers.

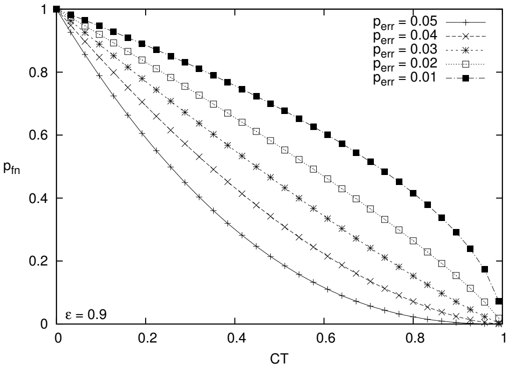

Fig. 10. Probability of False Negative vs. Compliance Threshold and $p_{\text{err}}$ for Detecting Opportunistic Attackers.

Our behavior rule based IDS allows one to adjust the minimum compliance degree threshold $C_T$ to obtain an acceptable $p_{\text{fn}}$ while keeping $p_{\text{fp}}$ as low as possible.

Figure 8 shows the relationship between $p_{\text{fn}}$ and $C_T$ for detecting a random attacker with varying $p_a$ values. Our intent is to analyze the effect of $p_a$ on $p_{\text{fn}}$. For each curve, $p_{\text{fn}} = 1$ at $C_T = 0$ and $p_{\text{fn}} = 0$ at $C_T = 1$. We see $p_{\text{fn}}$ decreases as $p_a$ increases because bad nodes are more likely to behave in a way that reveals their malintent.

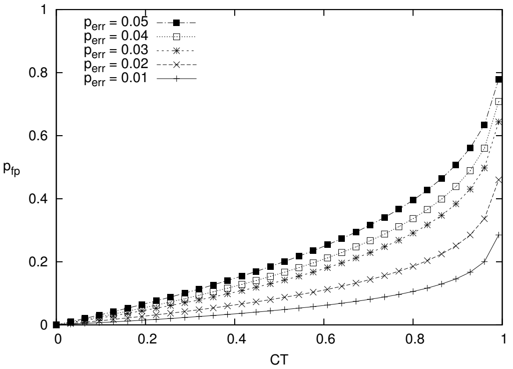

Fig. 11. Probability of False Positive vs. Compliance Threshold and $p_{\text{err}}$ for Detecting Good Nodes.

Figure 9 shows the relationship between $p_{\text{fn}}$ and $C_T$ for detecting a reckless attacker ($p_a = 1$) with varying $p_{\text{err}}$ values. Our intent is to analyze the effect of $p_{\text{err}}$ on $p_{\text{fn}}$. Like Figure 8, $p_{\text{fn}} = 1$ at $C_T = 0$ and $p_{\text{fn}} = 0$ at $C_T = 1$ for each curve. We see $p_{\text{fn}}$ decreases as $p_{\text{err}}$ decreases because noise is less likely to mask the malicious behavior of a reckless attacker.

Likewise, Figure 10 shows the relationship between $p_{\text{fn}}$ and $C_T$ for detecting an opportunistic attacker ($\varepsilon = 0.9$) with varying $p_{\text{err}}$ values. Like Figure 9, $p_{\text{fn}} = 1$ at $C_T = 0$ and $p_{\text{fn}} = 0$ at $C_T = 1$ for each curve. However, unlike Figure 9, we see $p_{\text{fn}}$ decreases as $p_{\text{err}}$ increases because an opportunistic attacker's attack probability ($p_a$) is higher (i.e., more aggressive) as noise is higher, thus increasing its probability of being detected and resulting in a smaller $p_{\text{fn}}$.

Correspondingly, Figure 11 shows the relationship between $p_{\text{fp}}$ and $C_T$ for detecting a good node with varying $p_{\text{err}}$ values. Our intent is to analyze the effect of $p_{\text{err}}$ on $p_{\text{fp}}$. For each curve, $p_{\text{fp}} = 0$ at $C_T = 0$. $p_{\text{fp}}$ decreases as $p_{\text{err}}$ decreases because noise is less likely to distort the behavior of good nodes to appear malicious.

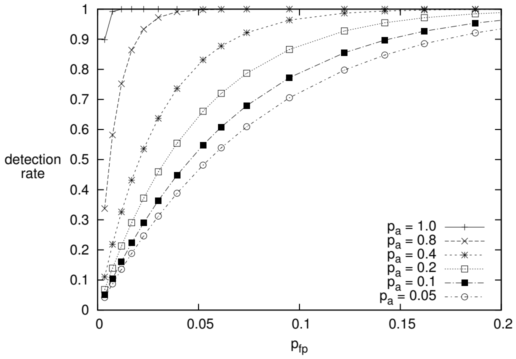

Fig. 12. ROC Graph under Binary Grading for Detecting Random Attackers.

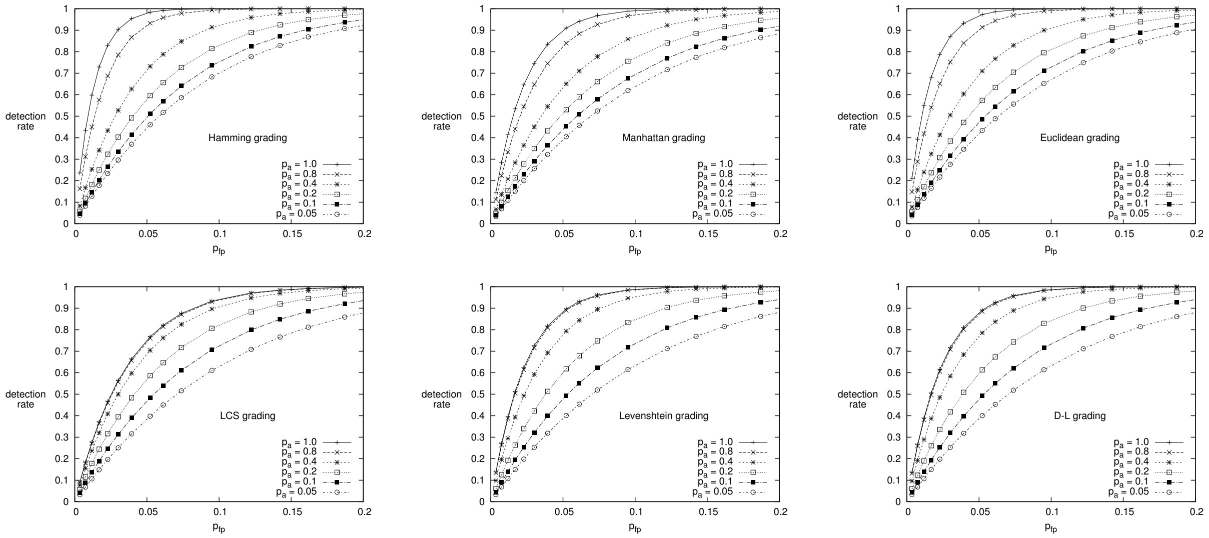

Fig. 13. ROC Graph under Each Distance-Based Grading Strategy for Detecting Random Attackers.

## V. Comparison of ROC under Binary and Distance-Based Grading Policies

By adjusting $C_T$, our specification-based IDS technique can effectively trade higher false positives off for lower false negatives to cope with more sophisticated and hidden random attackers. That is, by increasing $C_T$, one can effectively reduce $p_{\text{fn}}$ at the expense of $p_{\text{fp}}$. This is especially desirable for ultra safe and secure MCPS applications for which a false negative may have a dire consequence.

Figure 12 shows a ROC graph of intrusion detection rate $(1 - p_{\text{fn}})$ vs. false positive probability $(p_{\text{fp}})$ under the binary grading policy for reckless and random attackers, obtained as a result of adjusting $C_T$. In Figure 12 there are several curves, one for each random attacker case with a different attack probability $p_a$. We fix $p_{\text{err}}$ to 0.01 to isolate out its effect. As we increase $C_T$, the detection rate increases (vertically up on a ROC graph) while the false probability increases (toward the right of a ROC graph). We see that in this environment setting with our specification-based IDS technique, the detection rate of the VSM medical device can approach 100% for detecting attackers, that is, an attacker is always detected with probability 1 without false negatives, while bounding the false positive probability to below 5% (for reckless attackers) and 25% (for random attackers).

Figure 13 compares the performance of distance-based grading strategies for reckless and random attackers. The Area Under the Curve (AUC) is a common criterion for relating ROC graphs. Figure 13 shows that AUC increases as $p_a$ increases; each distance-based grading strategy performs better for more aggressive attackers. For reckless attackers $(p_a = 1)$, Hamming grading performs the best followed by Euclidean, Manhattan, Levenshtein, Damerau-Levenshtein and Longest Common Subsequence. However for random attackers, however, we see that Levenshtein grading performs the best followed by Damerau-Levenshtein, Hamming, Longest Common Subsequence, Euclidean and Manhattan.

## VI. Comparative Analysis

We perform a comparative study using the IDS design by Park et al. [21] and Tsang and Kwong [25] as baseline schemes. We only included these two studies in the comparative performance analysis because other studies did not provide adequate data.

### A. *Park et al. Study*

Park et al.'s IDS scheme is designed for detecting abnormal patient behaviors in a pervasive healthcare system. We justify our comparison between the abnormal behavior of Park's patients and our reckless adversary because of the way they synthesized their abnormal patients; time shifting data for normal patients is similar to a replay attack.

First, their IDS applies a similarity function, to grade four aspects of sensor data: longest common subsequence (LCS) of events $(s_1)$, number of common events that are not part of the LCS $(s_2)$, event start time similarities $(s_3)$ and event duration similarities $(s_4)$. They experiment with two variants each of $s_3$ and $s_4$: one considers events in the LCS ($s_D$ trials) and the other does not ($s_I$ trials). Their intent is to control the effects of interdependence between the similarity measures. Second, their IDS calculates a threshold for classifying good and bad behavior using a training data set. Third, the authors' IDS determines the weight for each of the four sensor data aspects. Park et al. measure patient activity using 3-tuple events, $e_i$, which comprise <sensor ID, time, duration>. They form episodes, $E_i$, from sequences of events. They use 70% of the dataset in [24] as normal training data and synthesize abnormal training data by random generation and time shifting normal training data by four, eight and 12 hours. They use the remaining 30% of the dataset as test data. Finally they optimize the performance by weighting the LCS $(s_1)$ and duration $(s_4)$ aspects of sensor data more heavily than non-LCS common events $(s_2)$ and start times $(s_3)$ aspects. Figure 14 shows the resulting ROC curves for several distinct configurations out of which we use the best ROC curve for performance comparison.

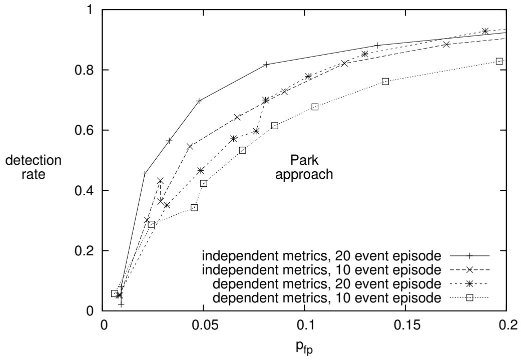

Fig. 14. ROC Graph for Park's IDS.

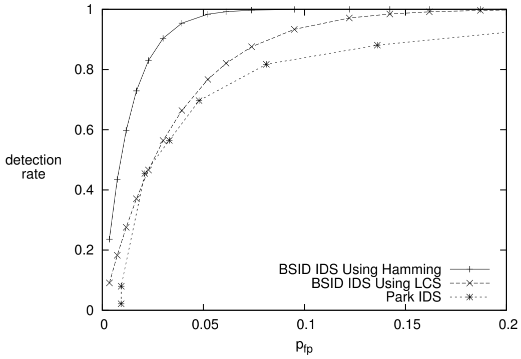

Fig. 15. Comparing ROC Graphs for Reckless Attacker Detection.

Figure 15 compares the performance of Park's design with our BSID design using the LCS grading strategy and a reckless attacker. BSID using the LCS grading strategy outperforms Park's IDS because the AUC of BSID using LCS dominantly covers that of Park's IDS. Given the same $p_{\text{fp}}$ of 5%, BSID has a better detection rate (98 versus 70%). Conversely, given the same detection rate of 90%, BSID has a better $p_{\text{fp}}$ (3 versus 14%). This could be due to our longer history; we use 100 events compared to 20. In addition to LCS ($s_1$), Park et al. consider three other measurements of audit data: number of common events that are not part of the LCS ($s_2$), event start time similarities ($s_3$) and event duration similarities ($s_4$). They use a weighting scheme that optimizes the performance, but our preliminary results indicate weights of 0 for $s_2$, $s_3$ and $s_4$ are best. This means one good measurement with a long history outperforms multiple optimally weighted measurements. In Figure 15, we also clearly see that BSID using the Hamming grading policy performs the best among all in detecting a reckless attacker.

### B. *Tsang and Kwong Study*

Tsang and Kwong report detection rates between 88.39 and 92.23% and false positive rates between 1.17 and 2.79% depending on the independent component analysis (ICA) technique used to prepare the audit data.

We modeled comparable ROC detection rates using reckless attackers. Table VI summarizes the comparison results. BSID performs better compared to Tsang and Kwong using the clustering algorithm yielding the highest detection rate. BSID produces a detection rate of 92.408% given a false positive rate of 0.666%. Using the FastICA clustering algorithm, their detection rate is only 92.23% and brings a higher false positive rate of 1.53%. If we allow a false positive rate as high as 1.53%, BSID can produce a detection rate as high as 99.742%.

TABLE VI
BSID ($p_{\text{err}} = 0.01$) VERSUS TSANG AND KWONG PERFORMANCE DATA

| Design | Detection Rate | $p_{\text{fp}}$ |
|---|---|---|
| BSID | 92.408% | 0.666% |
| BSID | 99.742% | 1.533% |
| Tsang and Kwong | 92.23% | 1.53% |

## VII. LESSONS LEARNED

We summarize lessons learned of applying the behavior rule specification-based intrusion detection (BSID) technique developed in the paper to medical devices, using VSMs as a running example. The first step is to specify the behavior rule set for a VSM as illustrated in Table I. The second step, based on the knowledge of environment noise represented by $p_{\text{err}}$, is to mechanically transform rules into state machines as illustrated in Figures 2 and 3 to differentiate good states from bad states. The third step, based on the knowledge of the attacker archetype, is to collect compliance degree data (Figures 4 - 7), parameterize the compliance degree distribution (Tables IV and V) and estimate $p_{\text{fn}}$ and $p_{\text{fp}}$ (Figures 8 - 11) from which the ROC graphs may be generated (Figure 12 under binary grading and Figure 13 under distance-based grading) for IDS performance assessment.

A key insight observed is that the accuracy of our IDS technique hinges on the completeness of the behavior rule set for specifying a VSM device since it is the very first step for defining acceptable or malicious behaviors. As behavior rules are derived directly from threats, the threat model must be broad enough to cover all possible threats that exploit system vulnerabilities. This places the responsibility for developing a complete attack model with the system designers. When a threat is overlooked, the state machine will lack unsafe states associated with the overlooked attack behavior indicator, and the attack will go undetected. Consequently, when new threats are discovered and introduced to the threat model, new behavior rules corresponding to the new threats must be added to the rule set because behavior rules are derived directly from threats. BSID allows newly identified threats to be updated to the threat model and hence the corresponding new behavior rules to be derived from which the state machine is automatically generated for intrusion detection. Another insight gained is that there is a tradeoff between $p_{\text{fn}}$ and $p_{\text{fp}}$, and this tradeoff is sensitive to the attacker archetype, namely, reckless, random and opportunistic, considered in the paper. Therefore, BSID, given the attacker archetype as input, can effectively identify the best tradeoff between $p_{\text{fn}}$ and $p_{\text{fp}}$ by setting the best $C_T$ value (through Equations 8 and 9) to satisfy the MCPS security requirement, such as minimizing $p_{\text{fn}}$ without violating the imposed threshold requirement for $p_{\text{fp}}$.

## VIII. CONCLUSIONS

For safety-critical MCPSs, being able to detect attackers while limiting the false alarm probability to protect the welfare of patients is of utmost importance. In this paper we proposed a behavior-rule specification-based IDS technique for intrusion detection of medical devices embedded in a MCPS. We exemplified the utility with VSMs and demonstrated that the detection probability of the medical device approaches one (that is, we can always catch the attacker without false negatives) while bounding the false alarm probability to below 5% for reckless attackers and below 25% for random and opportunistic attackers over a wide range of environment noise levels. Through a comparative analysis, we demonstrated that our behavior-rule specification-based IDS technique outperforms existing techniques [21], [25] based on anomaly intrusion detection.

In future work, we plan to analyze the overheads of our detection techniques such as the various distance-based methods in comparison with contemporary approaches. We also plan to deepen adversary modeling research based on stochastic Petri net techniques [10], [11], [19] such that the system can dynamically adjust $C_T$ to maximize intrusion detection performance in response to changing attacker behaviors at runtime.

## References

[1] H. Al-Hamadi and I. R. Chen. Redundancy management of multipath routing for intrusion tolerance in heterogeneous wireless sensor networks. *IEEE Transactions on Network and Service Management*, 10(2):189–203, 2013.

[2] M. Aldebert, M. Ivaldi, and C. Roucolle. Telecommunications Demand and Pricing Structure: An Econometric Analysis. *Telecommunication Systems*, 25:89–115, 2004.

[3] M. Anand, E. Cronin, M. Sherr, M. Blaze, Z. Ives, and I. Lee. Security challenges in next generation cyber physical systems. *Beyond SCADA: Networked Embedded Control for Cyber Physical Systems*, 2006.

[4] B. Asfaw, D. Bekele, B. Eshete, A. Villafiorita, and K. Weldemariam. Host-based anomaly detection for pervasive medical systems. In *Fifth International Conference on Risks and Security of Internet and Systems*, pages 1–8, October 2010.

[5] F. Bao, I. R. Chen, M. Chang, and J. H. Cho. Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection. *IEEE Transactions on Network and Service Management*, 9(2):169–183, 2012.

[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 Transactions on Industrial Informatics*, 7(2):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 *First Workshop on Cyber-physical Systems Security, DHS*, 2009.

[8] I. R. Chen and T. H. Hsi. Performance analysis of admission control algorithms based on reward optimization for real-time multimedia servers. *Performance Evaluation*, 33(2):89–112, 1998.

[9] I. R. Chen, A. P. Speer, and M. Eltoweissy. Adaptive fault tolerant qos control algorithms for maximizing system lifetime of query-based wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 8(2):161–176, 2011.

[10] I. R. Chen and D. C. Wang. Analysis of replicated data with repair dependency. *The Computer Journal*, 39(9):767–779, 1996.

[11] I. R. Chen and D. C. Wang. Analyzing Dynamic Voting using Petri Nets. In *15th IEEE Symposium on Reliable Distributed Systems*, pages 44–53, Niagara Falls, Canada, October 1996.

[12] S.-T. Cheng, C.-M. Chen, and I. R. Chen. Dynamic quota-based admission control with sub-rating in multimedia servers. *Multimedia Systems*, 8(2):83–91, 2000.

[13] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes. Using model-based intrusion detection for SCADA networks. In *SCADA Security Scientific Symposium*, pages 127–134, Miami, FL, USA, January 2007.

[14] A. daSilva et al. Decentralized intrusion detection in wireless sensor networks. In *1st ACM inter. workshop on quality of service & security in wireless and mobile networks*, pages 16–23, 2005.

[15] B. Dutertre. Formal modeling and analysis of the modbus protocol. *Critical Infrastructure Protection*, pages 189–204, 2007.

[16] C. Hsu. Many popular medical devices may be vulnerable to cyber attacks. http://www.medicaldaily.com/news/20120410/9486/medical-implants-pacemaker-hackers-cyber-attack-fda.htm, April 2012.

[17] K. Ioannis, T. Dimitriou, and F. Freiling. Towards intrusion detection in wireless sensor networks. In *13th European Wireless Conference*, 2007.

[18] I. Lee and O. Sokolsky. Medical cyber physical systems. In *47th ACM Design Automation Conference*, pages 743–748, 2010.

[19] Y. Li and I. R. Chen. Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks. *IEEE Transactions on Mobile Computing*, 10(3):349–361, 2011.

[20] R. Mitchell and I. R. Chen. Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems. *IEEE Transactions on Reliability*, 62(1):199–210, March 2013.

[21] K. Park, Y. Lin, V. Metsis, Z. Le, and F. Makedon. Abnormal human behavioral pattern detection in assisted living environments. In *3rd ACM International Conference on Pervasive Technologies Related to Assistive Environments*, pages 9:1–9:8, 2010.

[22] P. Porras and P. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In *20th National Information Systems Security Conference*, pages 353–365, 1997.

[23] S. M. Ross. *Introduction to Probability Models, 10th Edition*. Academic Press, 2009.

[24] E. Tapia, S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In A. Ferscha and F. Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 158–175. Springer Berlin / Heidelberg, 2004.

[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 *IEEE International Conference on Industrial Technology, 2005.*, pages 51–56, December 2005.

[26] K. Venkatasubramanian and S. K. S. Gupta. *Security in distributed, grid, mobile, and pervasive computing*, chapter Security Solutions for Pervasive Healthcare. Auerbach Publications, 2007.

[27] W.-S. Yang and S.-Y. Hwang. A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*, 31(1):56–68, 2006.

[28] O. Yilmaz and I. R. Chen. Utilizing call admission control for pricing optimization of multiple service classes in wireless cellular networks. *Computer Communications*, 32(2):317–323, 2009.

[29] C. Zimmer, B. Bhat, F. Mueller, and S. Mohan. Time-based intrusion detection in cyber-physical systems. In *1st ACM/IEEE International Conference on Cyber-Physical Systems*, pages 109–118, 2010.

**Robert Mitchell** received the BS, MS and PhD degrees in Computer Science from Virginia Tech in 1997, 1998 and 2013, respectively. He works at The Boeing Company. His research interests include intrusion detection, cyber-physical systems, security and modeling and simulation.

**Ing-Ray Chen** received the BS degree from the National Taiwan University, Taipei, Taiwan, and the MS and PhD degrees in Computer Science from the University of Houston. He is a professor in the Department of Computer Science at Virginia Tech. 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, The Computer Journal, Wireless Personal Communications, and Security and Communication Networks. He is a member of the IEEE and ACM.