Understanding the Security of Interoperable Medical Devices using Attack Graphs

Curtis R. Taylor, Krishna Venkatasubramanian, Craig A. Shue
Department of Computer Science,
Worcester Polytechnic Institute
Worcester, MA, 01609
crtaylor@cs.wpi.edu, kven@wpi.edu, cshue@cs.wpi.edu

ABSTRACT

Medical device interoperability is an increasingly prevalent example of how computing and information technology will revolutionize and streamline medical care. The overarching goal of interoperable medical devices (IMDs) is increased safety, usability, decision support, and a decrease in false alarms and clinician cognitive workload. One aspect that has not been considered thus far is ensuring IMDs do not inadvertently harm patients in the presence of malicious adversaries. Security for medical devices has gained some traction in the recent years following some well-publicized attacks on individual devices, such as pacemakers and insulin pumps. This has resulted in solutions being proposed for securing these devices, usually in stand-alone mode. However, the introduction of interoperability makes medical devices increasingly connected and dependent on each other. Therefore, security attacks on IMDs becomes easier to mount in a stealthy manner with potentially devastating consequences.

This work outlines our effort in understanding the threats faced by IMDs, an important first step in eventually designing secure interoperability architectures. In this regard, we present: (1) a detailed attack graph-based analysis of threats on a specific interoperability environment based on providing a patient pain medication (PCA), under various levels of interoperability from simple data aggregation to fully closed-loop control; (2) a description of the mitigation approaches possible for each of class of attack vectors identified; and (3) lessons learned from this experience which can be leveraged for improving existing IMD architectures from a security point-of-view. Our analysis demonstrates that *even if we use provably safe medical systems in an interoperable setting with a safe interoperability engine, the presence of malicious behavior may render the entire setup unsafe for the patients, unless security is explicitly considered.*

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General - Data communications, Security and protection; K.6.5 [**Computing Milieux**]: Security and Protection - Authentication, Unauthorized access; J.3 [**Computer Applications**]: Life and Medical Sciences - Medical information systems

General Terms

Security, Interoperability, Medical Device Systems

Keywords

Interoperable Medical Devices, Security, PCA, Infusion Pump

HiCoNS'14, April 15–17, 2014, Berlin, Germany.

http://dx.doi.org/10.1145/2566468.2566482 .

1. INTRODUCTION

Medical systems are increasingly being connected to each other as a way to improve patient safety [26]. The ability of medical devices to interoperate with one another has the potential to yield better performance, from reduced false alarms to automatic decision/diagnosis support and medication interaction checking in real-time [1]. Not surprisingly, interoperability has been predicted to improve patient outcomes by reducing the 95K - 195K errors committed in U.S. hospitals [12].

While there may be impediments to device manufacturers providing interoperability with their competitors' medical devices, such as a lack of data standards, alternative mechanisms are possible. In particular, a communication/middleware standard would allow heterogenous devices to communicate with one another. The *Medical Device Plug-n-Play Integrated Clinical Architecture* (ICE) is a result of such standardizing efforts [2]. Although there can be interoperability at many different granularities from technical (being able to exchange bytes) to conceptual (shared assumptions about the reality at a meaningful abstraction) [28, 31], the interoperability in the ICE standard is somewhere in between syntactic (data format of communication is standard) and semantic (the meaning of the data being exchanged is unambiguously defined) interoperability.

The goal of ICE is to enable *safe interoperability* between medical devices. Specifically, safety in this context is defined as ensuring the patient's health is not harmed in anyway by the use of the medical devices in an interoperable fashion. One important issue that is not addressed in this standard is security. Considering security for IMDs is necessary because: (1) they deal with sensitive patient information, (2) laws such as the Health Insurance Portability and Accountability Act (HIPAA) [14], mandate it, and (3) security attacks

can have serious safety consequences for the patients. In particular, a malicious entity can now easily suppress legitimate information and introduce bogus information between the devices and the middleware, leading to untimely or unwanted actuation or loss of privacy. *Therefore, we contend that both security and safety need to be enforced in IMDs to ensure that the patient's health outcomes are not worsened under any circumstances.* Recent years have brought increased attention to security vulnerabilities in standalone medical devices [7,8,15,17]. However, the introduction of interoperability makes medical devices increasingly connected and dependent on each other. Therefore, security attacks on IMDs becomes easier to mount, and in a stealthy manner.

There has been growing interest in security issues pertaining to medical data collection, data transfer and processing, and electronic medical health records [5, 13, 30]. Standardization efforts are also underway [6, 20, 21]. In [11], the authors performed a detailed survey of existing communication and data standards in the medical domain and the techniques they deploy for security purposes which can be used for medical device interoperability purposes. It was found that significant gaps existed in the today's standards in terms of security particularly relating to communication security. This myopic focus on safety without considering the whole spectrum of security issues makes these standardization efforts essentially incomplete. The proper development of strong security solutions for IMDs is still an open research question.

To develop security solutions for IMDs, we need a good understanding of the various threats to an IMD setup. Instead of analyzing the security requirements of IMDs as a whole as done by earlier efforts [35] [32], which forces one to abstract out situation specific details and therefore make very broad conclusions, we take bottom-up approach in this paper. We present a detailed description of attacks on a specific interoperability scenario for patient controlled analgesia (PCA). This *PCA-IMD* setup consists of a PCA pump, and pulse-oximeter (measures O_2 levels in the blood) and a capnograph (measures CO_2 levels in the blood) and the goal is to allow patents to infuse pain medication as needed without over-infusing indicated by onset of respiratory depression. We further consider various levels of interoperability for this PCA-IMD scenario from simple-cases where interoperability promotes data aggregation to fully-closed-loop control of all three medical devices. The *principal contributions* of this paper therefore are:

- An attack graph-based description of attacks on IMDs when considering the PCA-IMD interoperability scenario.
- A description of the general mitigation strategies for each class of the attacks that are possible on the IMDs.
- A description of lessons learned from our experience, which can be used to design the interoperability architecture in a security-conscious manner.

We chose this PCA scenario for our IMD case-study because it is responsible for a very large number of treatment errors in the hospital setting. One study estimated that there are anywhere between 600,000 to 2 million adverse events in U.S. hospitals every year related to PCA [27]. Our analysis demonstrates that even if we use provably safe medical systems in an interoperable setting with a safe interoperability middleware, the presence of malicious behavior renders the entire setup unsafe — potentially harm inducing — for the patients.

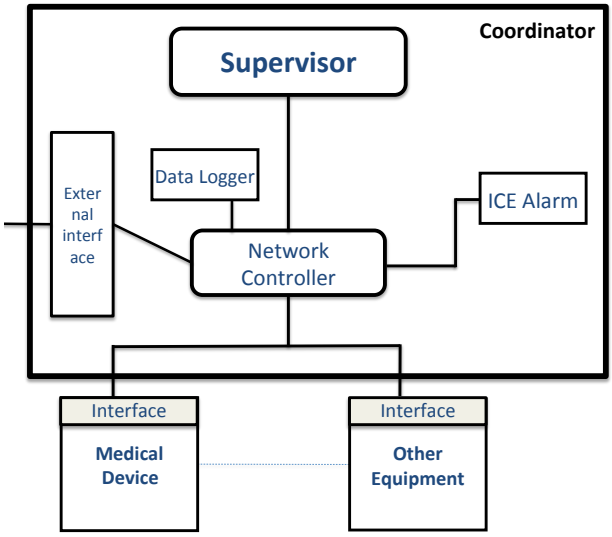

Figure 1: Interoperability architecture of MD PnP ICE standard

To the best of our knowledge this is the first systematic attack description for a common treatment scenario (i.e., pain management), which can be implemented with IMDs in a realistic setting. Our work demonstrates that security has profound consequences to the safety of medical device interoperability and the patients they are serving. It is not just enough to design IMDs to be able to handle device failures and communication and software errors in order to be safe. They have to be secured from a variety of malicious behavior as well to be truly safe.

The paper is organized as follows: Section 2 presents background information on interoperability architecture standards and potential deployment approaches. Section 3 presents our problem statement along with the system and trust model. Section 4 illustrates attacks on the system. Section 5 presents the lessons learned and Section 6 presents the related work. Finally, Section 7 concludes the paper and presents the future work.

2. BACKGROUND

Rather than wait for the medical devices from different manufacturers to organically evolve interoperability capabilities, the *Integrated Clinical Architecture* (ICE) was created to enable diverse devices to talk to one another [2]. ICE was designed to act as a middleware to enable interaction of legacy, stand-alone medical devices and the applications using the medical devices. It has the potential to provide anything from data aggregation to closed-loop control over the patient's health. The architecture of ICE typically consists of three entities (see Figure 1):

- A collection of *Medical Devices* on or around a single patient that can perform monitoring and actuation.
- The *Supervisor* receives data from the various medical devices, processes it, and initiates action from the medical devices. The Supervisor runs clinical applications

(referred to as *apps* from now on) that use the connected devices to support a clinical scenario selected by the caregiver.

- The *Network Controller* interfaces with one or more medical devices and the supervisor. It is responsible for collecting data from the individual devices. It also connects the entire setup to an external network, such as the Healthcare Information System (HIS). The network controller also records all the actions of the entire system in a data logger for future analysis.

IMDs are configured for each patient according to their individual needs. The caregiver is responsible for configuring the IMDs, which means: (1) identifying the medial devices that are needed to monitor or treat the patient, (2) connecting the devices to the network controller and the supervisor, (3) selecting an appropriate app on the Supervisor for enabling interoperability, and (4) monitoring the patient's well-being through the Supervisor. The caregiver can control various parameters of the system, such as alarm thresholds or algorithms for performing closed-loop control of patients, all through the apps running on the supervisor.

3. PROBLEM STATEMENT

Solutions for building safe IMDs only considers "naturally-occurring" faults within the system. These do not include faults introduced into the system by adversaries, which may not follow the models of "naturally-occurring" faults, but instead act in unexpected ways. Hence, analyzing the security threats for interoperable medical devices is very important for ensuring that the IMDs are safe and do not harm the patient.

In this paper, we investigate the various ways in which a specific instantiation of interoperable medical devices can be attacked, in a systematic manner. We specifically consider cases where the individual devices are themselves "correct-by-design" and therefore are considered "safe" when they are operating in stand-alone fashion [22] . However, when malicious behavior is allowed, even such provably-safe devices working in conjunction with a safe and trusted coordinator in an interoperable environment, are inherently unsafe. We consider this analysis as a step towards building an effective architecture for secure interoperable medical devices that expands on the ICE standard. Before delving into the details of our security analysis, we present our system model and trust/threat model for this work.

3.1 System Model

The ICE standard for interoperability between medical devices can support any combination of medical devices, provided they can be coordinated in a meaningful way to provide effective care for patients [2]. The IMD configuration will vary to account for each patient's specific situation. In order to understand the security threats on IMDs, we consider a small IMD system, consisting of three devices, for a single patient needing pain management. As we will see, even in this very limited scenario, the avenues of attack are large and we can draw broad conclusions about security threats to IMDs in general.

Our scenario, referred to as *PCA-IMD*, consists of an infusion pump programmed to infuse pain medication (e.g., morphine) to the patient at a specific (basal) rate in a hospital or care-facility. As pain medications tend to suppress respiration, we also have a pulse-oximeter (measures level of O_2 in the blood) and a capnograph (measures level of CO_2 in the blood) to determine how the patient is responding to the pain medication. The pulse-oximeter and the capnograph are collectively referred to as *sensors*, in the rest of the paper. The infusion pump also allows the patient to press a bolus button to receive a single, large dose of the medication as needed. Obviously, frequent boluses should only be allowed for a patient if it is not suppressing their respiration to unhealthy levels.

All the medical devices in our setup interact with the coordinator. The details of the coordinator entity are abstracted out as our focus is primarily on its interaction with the medical devices. The coordinator is programmed by the caregiver by loading *medical applications* on it that perform specific tasks such as alarming or providing closed-loop control. In many instances, the coordinator can be used to control the individual medical devices. The coordinator has an internal alarm and logging capability and is connected to a patient display, which displays the patient's status in terms of physiological signals (O_2 and CO_2 in our case) trends. The caregiver essentially monitors the patient through the patient display (dashed arrow in Figure 2). The coordinator also interfaces with the hospital electronic health record (EHR) system. It can update and query the EHR when needed. For example, a medical application running on the coordinator can be used to perform a sanity check on the nurse's programming of the infusion pump based on medication orders in the EHR.

Our interoperability setup can be classified into four *configurations* based on the level of control associated with the coordinator:

- *Simple (SC):* With a simple coordinator, the infusion pump and the sensors are programmed directly by the caregiver and then connected to the coordinator. The coordinator receives status updates from the individual medical devices, and it displays the information to the caregiver via the patient display. If the blood oxygen level of the patient goes below a certain threshold, a medical application on the coordinator will raise an alarm to the caregivers.

- *Alarming (AC):* In this scenario, the coordinator has the capability to program the devices as specified by the caregiver and monitor the patient's condition. If the blood oxygen level goes below a certain threshold, the coordinator (through a medical application executing on it) raises an alarm for the caregivers to react.

- *Bolus-controlling (BC):* In this scenario, the coordinator has the capability to program the devices as specified by the caregiver, raise an alarm if the patient's condition deteriorates, and control the frequency with which the patient can give themselves bolus doses.

- *Fully-closed Loop (FC):* In this scenario, the coordinator, after the initial medical device programming by the caregiver, monitors the patient's condition, and if it deteriorates, automatically modifies the programming in place to reduce the safety risk, such as over-infusion, to the patient. Further, it can raise an alarm for the caregivers and also control the bolus volume and frequency for the patients.

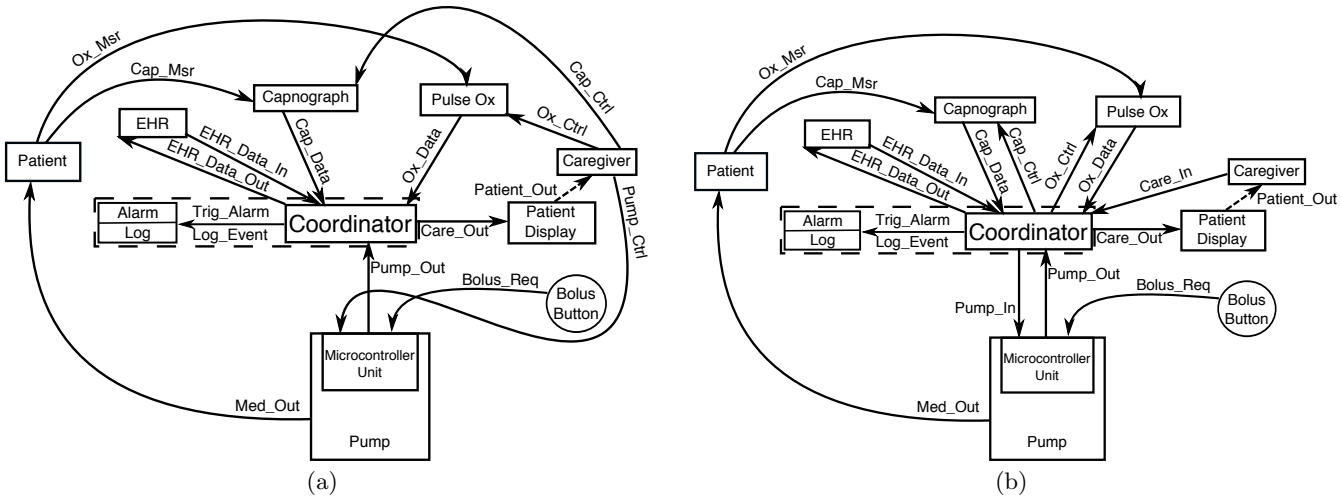

Figure 2: System Model for Interoperability Threat Analysis (a) Simple Case (SC), (b) Other cases (AC,BC, and FC)

Figure 2 (a), (b) shows the assumed interoperability setup for SC and the other modes (AC, BC and FC), respectively. The edges are labeled to indicate the information exchanged between the entities that the edge connects.

3.2 Trust Model

In our interoperability scenario, we consider the coordinator and the associated logging and alarms to be the only members of the trusted computing base (TCB). These components are trusted (they do not have malicious intent) and trustworthy (they will operate as expected). The dashed box in Figure 2 (a) and (b) signifies the TCB in our system model. Further, we assume that the caregiver is not necessarily trustworthy, in that the caregiver can make mistakes in programming the devices, but does not have malicious intent. We further assume that the infusion pump in our system model is verifiably safe as described in [22].

For our work, we essentially consider active adversaries (also called "attackers" interchangeably) who may interfere with communication links, as per the Yao-Dolev model of an adversary [9]. In addition, the adversary may also physically alter the infusion pump or the line from the infusion pump to the patient, the coordinator, the pulse oximeter and capnograph. We assume that adversaries cannot modify the firmware of the devices, but they can mount limited physical attacks on the IMD setup. For example, the attacker can induce readings in the sensor and cut the infusion line to the patient. Note that, while adversaries may simply inject the patient directly and induce a medical emergency, we consider such attacks outside the scope of interoperable medical device security.

Finally, we only consider adversaries that induce over-infusion (for pain medication under-infusion does not hamper patient safety) through the infusion pump. In other types of interoperability scenarios, both under-infusion and over-infusion can be problematic, such as with insulin infusion. This essentially doubles the threat surface.

4. ANALYSIS OF ATTACKS ON PCA-IMD SCENARIO

Before we can understand the security of IMDs, we must first examine their attack surfaces and the associated vulnerabilities. Importantly, by focusing on the assets to be protected and their associated vulnerabilities, we can determine remediation opportunities without having to anticipate an attacker's actions. In the context of medical devices, safety and security have a special relationship. The high-level patient safety goals vary dramatically based on a given patient scenario and set of devices. We therefore take a common treatment option in a hospital which can be improved using IMDs, evaluate its security in a systematic manner, and develop generalizable requirements for improving the safe operation of IMDs.

We consider an IMD scenario for patient-controlled analgesia (PCA), involving a PCA infusion pump, a pulse oximeter, and a capnograph, as a motivating example. In our scenario, there is one simple safety goal for the PCA pump: it must *not* administer an excessive quantity of pain medication (i.e., *over-infusion*). If this safety goal is violated, the patient's respiration may be suppressed and if not remediated, this may lead to patient mortality. In the remainder of this section, we focus on the attack vectors adversaries can use to subvert patient safety and harm the patient. We then discuss some viable countermeasures for these attacks.

4.1 Attack Graphs

The goal of the attack for the PCA-IMD scenario is to harm the patient by infusing excessive pain medication. Therefore over-infusion at the PCA pump is the only "unsafe" state for our case-study. If the infusion pump in our setup fails to infuse a sufficient quantity of analgesia, it is unlikely to cause a life-threatening event. Instead, the patient will experience pain and will alert a caregiver manually. When considering the safety and security of IMDs, each unsafe state must be identified and the paths to that unsafe state enumerated. In Figures 3, 4, 5 we depict the *attack graphs* that describe various *attack vectors* that can lead to the over-infusion state for our setup. *Each of these figures can be thought of as sub-branches of a larger attack graph for*

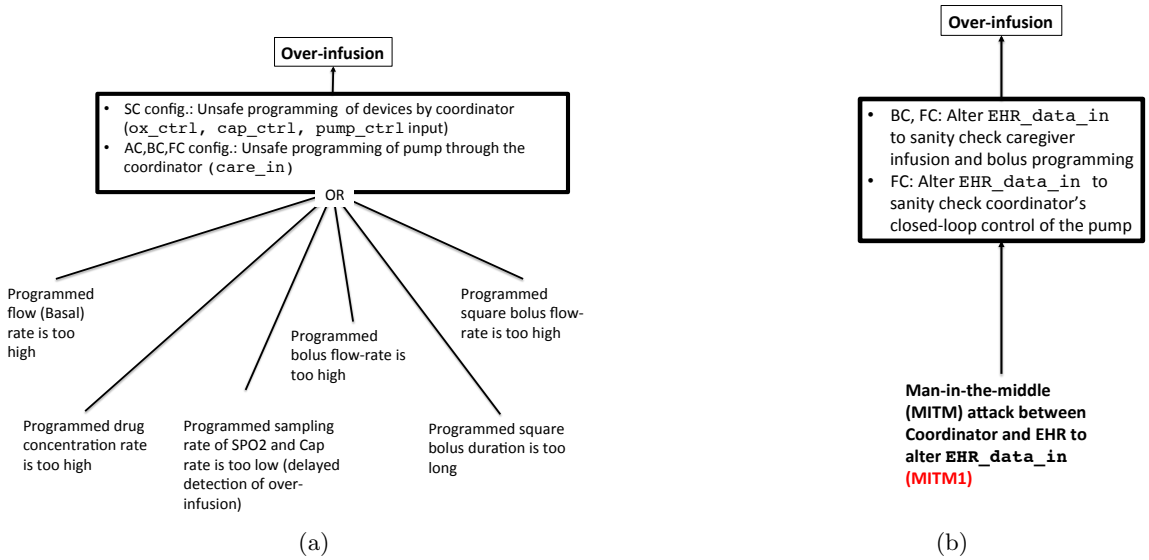

Figure 3: Attack Graphs representing attack vectors due to: (a) IMD initialization attacks, (b) EHR access attacks. Step 2 is referenced in the example attack in Section 4.2.1.

PCA-IMD. The figures are representing the following attack scenarios:

- *Initialization Attacks:* Represented in Figure 3 (a), these attacks represent the situations where the caregiver programs the devices (using `cap_ctrl`, `ox_ctrl`, `pump_ctrl` in the SC case, and using `care_in` through the coordinator in the AC, BC, and FC cases) incorrectly.

- *EHR Access Attacks:* Represented in Figure 3 (b), this attack represents the situations where the communication link between the coordinator and the EHR is compromised primarily through a man-in-the-middle attack.

- *Partial Feedback Attacks:* Represented in Figure 4 (a), these attacks represent the situations where the some of feedback channels to the coordinator (e.g, `pump_out`, `ox_data`, etc.) are rendered non-functional. Given that partial or lack of information from the devices, these attacks are probably the easiest to detect and raise alarms for. However, incomplete information at the coordinator may lead to incorrect decisions, especially in emergency situations where action needs to be taken in a time-sensitive manner.

- *Incorrect Feedback Attacks:* Represented in Figure 4 (b), these attacks represent the cases where the feedback received at the coordinator has incorrect information as a result of an adversary tampering with it. This can lead to wrong diagnosis, missed alarms and, in the FC configuration, incorrect actuation leading to over infusion.

- *Delayed Feedback Attacks:* Represented in Figure 5, these attacks include the cases where the feedback received at the coordinator is delayed as a result of an adversary. Such delayed feedback information may be interpreted as a current reading, causing over-infusion.

In last four attack graphs (involving feedback), we only show the avenues for attacks that can cause manipulation of the feedback to the coordinator. We do not attempt to describe the mechanisms for an attacker to perform such manipulation, since attempts to predict adversary behavior often lead to inadequate defenses. Instead, we focus on the broad outcomes of these attacks. Fortunately, many of the attacks can be thwarted with known countermeasures obtained from best practices in network security, software validation, and operating system security to ensure the attack cannot occur. However, one must be aware that attack vectors can be activated simultaneously by the attackers.

Broadly speaking all these attacks are manifestations of the *confused deputy* attack [18]. In a confused deputy attack, a privileged entity (the "deputy") is manipulated by an attacker to perform an unsafe act. Depending on the attack scenario, the caregiver, the coordinator, and the pump can be victims of a confused deputy attack. While the exact details vary for each entity, the general pattern is the same: the attacker would block, alter, or delay the information the deputy requires for proper operation. This would cause the deputy to make a medical decision with inaccurate or limited information. As an example, we consider a confused deputy attack on the caregiver. If the attacker wants to manipulate the caregiver into over-infusing the patient with pain medication, the attacker may alter the sensor readings from the pulse oximeter and the capnograph. In particular, the attacker may alter both sensor readings to indicate the patient's respiration is normal or elevated, regardless of the patient's actual respiration behavior. Accordingly, the caregiver may believe it is safe to administer a greater quantity of medication than what the patient can handle. If the attacker continues to report healthy readings, despite suppressed respiration, the attacker may manipulate the caregiver into programming a larger dose of medication when it is unsafe to do so.

As the model changes to have greater coordinator involvement, the attack vectors shift. Once the coordinator has the

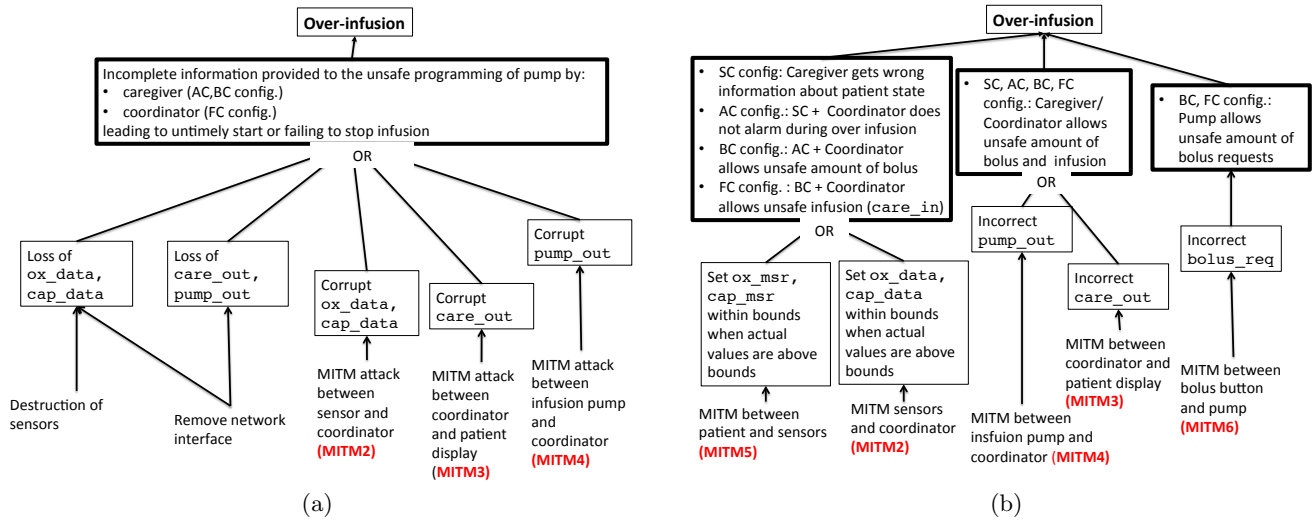

Figure 4: Attack Graphs representing: (a) partial feedback attacks, (b) incorrect feedback attacks. Steps 1 and 2 are referenced in the example attack in Section 4.2.1.

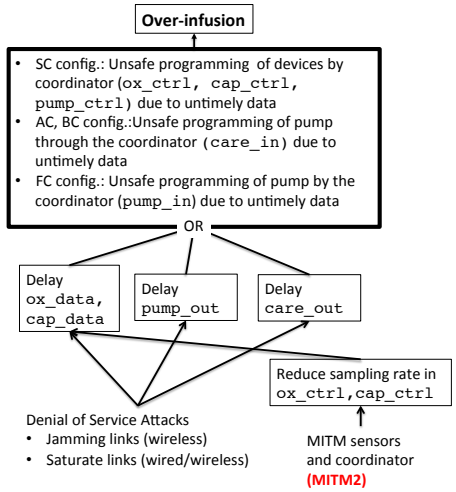

Figure 5: Attack Graphs representing delayed feedback attacks

responsibility of controlling boluses, an attacker can begin to manipulate the inputs to the coordinator with the goal of encouraging the coordinator to allow a bolus that it should prevent. In the FC configuration, the role of the caregiver is completely removed, placing these responsibilities on the coordinator. The essential issue in the FC configuration is to design closed-loop control of the coordinator application to be safe from causing the patient harm. While the change in the FC configuration may seem to introduce a security risk, the attack vectors remain largely the same. The only difference is that the attacker must focus on manipulating the coordinator instead of the caregiver.

4.2 Man-in-the-Middle Attack

The attack graphs shown in Figures 3(b), 4, and 5 demonstrate that one of the most common strategies for an attacker is to insert itself between two legitimate entities in the interoperability setup, i.e., cause Man-in-the-middle (MITM) attacks. The MITM attack can, in theory, be mounted between any two legitimate entities in the PCA-IMD. As shown in the Figures 3(b), 4, and 5, there are six such MITM scenarios ($MITM_1$,...,$MITM_6$). Note that we use MITM attacks as a way to illustrate the most general form of spoofing attacks (either through communication of physical compromise) in our system. The rest of this section is dedicated to showing how an attacker could go about mounting an MITM on PCA-IMD in a care-facility setting.

Figure 2 shows an example medical device interoperability setup within (a partial view of) a modern care facility. In this setup, each patient has a set of a PCA-IMD *apparatus* for regular monitoring and actuating treatment for pain management. All three of the medical devices in the PCA-IMD apparatus are connected to the hospital network. The pump through the wired network over $Switch_3$ and the pulse-ox and capnograph over the wireless network through the local access point. In addition, the apparatus has a wireless patient display, which is used by caregivers to view the patient's current health status, as well as access the EHR. The wireless network within the hospital may be used by the caregivers and visitors to access the various systems within the care facility network or access the Internet.

For simplicity of management, we assume each PCA-IMD apparatus has an individual coordinator, which is managed centrally within the care facility. When a patient is brought into the facility, the initial interoperability configuration information is then passed to a dynamically instantiated coordinator. This instantiation occurs on a per patient basis. Figure 6 shows the coordinator is connected to the network over $Switch_2$. For brevity, only one coordinator and simple network paths are shown. In an actual deployment, redundant network paths as well as multiple coordinators may be required.

Example 1: This pertains to the $MITM_2$ and the $MITM_3$ cases. The adversary broadcasts an SSID the same as that of the hospitals AP [4]. This will cause the wireless entities (pulse-ox and the capnograph in $MITM_2$ and patient display in $MITM_3$) in the interoperability apparatus to re-associate to the faux AP due to a higher RSSI value. The attacker then intercepts and forwards all communication from wireless entities to the hospital's AP, effectively becoming the man-in-the-middle. In the SC and AC configuration,

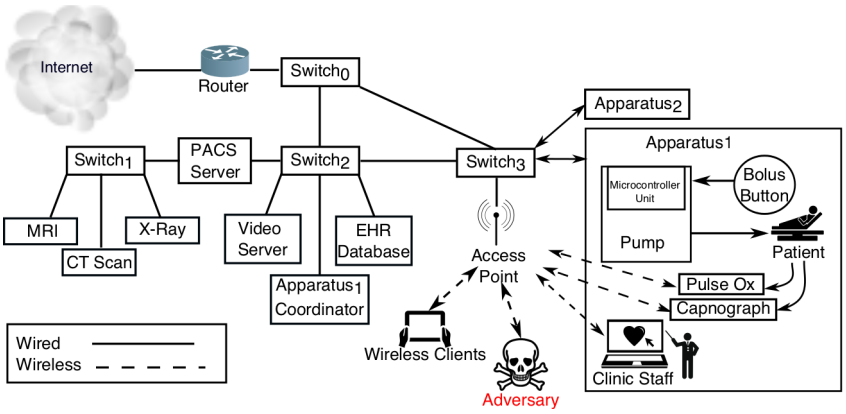

Figure 6: An example interoperable scenario in a care facility. Each patient's apparatus is given an individual coordinator. Some devices within the patient's room connect wirelessly while others are hardwired. The clinic staff uses the patient display to configure the coordinator and retrieve or update EHR records.

such an attack can be mounted to suppress local device alarms and coordinator alarms respectively.

Example 2: This pertains to the $MITM_1$ and the $MITM_4$ cases and requires the adversary to mount MITM on wired links between the EHR and the coordinator and the infusion pump and the coordinator, respectively. This is considerably more difficult as it requires physical access to the care facility's networking infrastructure, such as $Switch_2$. However, once such an access is available, then enabling MITM may be as simple as mounting an ARP poisoning attack [16], where the physical (LAN) address of the communicating entities is modified to that of the attacker during initial discovery using the Address Resolution Protocol (ARP).

Example 3: This pertains to the the $MITM_5$ and the $MITM_6$ cases, where MITM is mounted through physical compromise rather than by manipulating the communication between the entities. $MITM_5$ requires the modification of the physical sensor itself so that the actual patient state is not captured accurately. Similarly, $MITM_6$ requires physical modification of the infusion pump to be able to tamper with the bolus request information being sent from the bolus button.

Note that in the above analysis, we have assumed the infusion pump to be implemented correctly without interface or software defects. The attack example above is being described in a relatively error-free scenario. In reality, adverse events as a result of user interface issues and software defects occurred over 56,000 times from 2005-2009 [10]. Most of these were eventually detected after the devices monitoring the patient start reading irregular values. With MITM attacks, however, these devices cannot be trusted to be accurately relaying readings the coordinator. As a result, the actual scale of the security issue described in the paper is quite a bit larger.

4.3 Mitigating the Attacks

For over-infusion to occur, the infusion pump has to administer large quantities of pain medication in an untimely manner. There are four methods for an attacker to cause the controller to send the pump commands that trigger an over-infusion event are:

Programming-Focused: In this case, the caregiver's input is incorrect. The caregiver is not in our TCB and therefore can provide incorrect input to the devices (for SC) or the coordinator (AC and BC) either simply due to human error or incompetence. The caregiver can press incorrect keys when entering values, calculate rates incorrectly, or simply program the pump accurately, but use an incorrect concentration of the medication. **Mitigation:** These cases can be remediated using local solutions at the pump itself such as drug libraries, flow sensors, and barcode scanners [22]. One can push the remediation to the coordinator as well, but that would significantly increase the complexity of the coordinator, which is undesirable.

Communication-Focused: The inputs to the pump, `pump_in` and `bolus_req`, for the AC, BC and FC configurations, is incorrect. This is possible because: (a) some or all the information going out of the coordinator to the pump over `pump_in` has been altered (delayed, modified, corrupted) by adversaries, (b) bolus information going from the bolus button to the infusion pump over `bolus_req` has been altered (delayed, modified, corrupted) by adversaries; (c) some or all the information going into the coordinator from the sensors (i.e., `ox_data`, and `cap_data`), EHR (i.e., `EHR_data_in`), and the pump (i.e., `pump_out`) has been altered (delayed, modified, corrupted) by adversaries; and (d) the programming instructions from the caregiver to the coordinator, `care_in` has been altered (delayed, modified, corrupted). **Mitigation:** These can be prevented by using cryptographic primitives to preserve the confidentiality, integrity and authenticity properties of the lines of communication. Such techniques are considered best practices for securing network communication.

Hybrid : The caregiver's programming of the coordinator, `care_in`, in the AC and BC and FC configurations, is incorrect. **Mitigation:** All the reasons listed for the two aforementioned cases may apply and the same prevention strategies can be used.

Entity-Focused: If the the pump or the sensors or their environment are tampered with by adversaries, it is possible for the coordinator to be unaware of the actual state of the patient leading to over-infusion. **Mitigation:** In such cases, attack prevention (as in the three aforementioned cases) becomes very difficult. The only option is to detect problems with the patient's health based on data from the sensors and raise an alarm. However, if the sensors are not report-

ing correct data, the system simply lacks sufficient data to raise an alarm. The only way to deal with this situation is through redundancy of medical devices, assuming at least some of them are not compromised.

In summary, these vectors characterize the varied types of misinformation that could reach the PCA pump, the coordinator, and the caregiver. Within each vector, the attacker can devise a variety of actual attacks. The context of the IMD deployment plays a big role in identifying them. Any mitigation solution for these attacks have to therefore consider all of these cases.

4.4 Cryptographic Solutions

Several of the mitigation strategies rely on the use of cryptography, especially as a way to avoid MITM attacks. However, the use of cryptography is not without its problems.

- Medical devices typically do not support cryptographic operations, which may limit the deployability of the device. Cost in terms of their correct implementation, computational complexity and supporting infrastructure (e.g., certification authorities) is not non-trivial.

- Cryptography often relies on effective key distribution to work. Secure key distribution in a dynamic environment such as a hospital where the same device can be associated with multiple patients over a short span of time, is notoriously difficult. Approaches that are based on physiological signals [33] [3] may be applicable here, but they require diversity of signals which is not always available.

- When a new device is added to the interoperability setup, another concern would be if the device uses a protocol that relies on a *leap-of-faith (LoF)* mechanism. LoF mechanisms are those protocols in which the very first interaction between two parties assumes complete trust and results in the exchange of cryptographic primitives. All subsequent interactions then use this exchanged primitive for security [29]. This concern noticeably increases when considering the dynamic nature of IMDs and care facility workflows. Devices used for monitoring patients are continuously being added, removed and exchanged between different patients. As a consequence of this fluidity, devices will need to re-associate with network and re-establish connections leaving a space for potential vulnerabilities.

5. LESSONS LEARNED

The attack vectors in Figures 3, 4, and 5 highlight several important points:

- **Individual medical device safety does not equate to interoperability safety.** A device can be formally defined as "safe" if and only if none of its execution paths invoke a particular set of negative actions [22]. However, the safety of a particular medical device and the coordinator are insufficient to ensure that it remains safe in an interoperable setting. In our system model, adversary induced misinformation or bad input can cause an infusion pump to over-infuse medication, endangering patient safety. This condition can occur even if the infusion pump is guaranteed to meet its own safety requirements.

- **Secure communication within the IMD setup is paramount** As we transition from SC all the way to FC we can see that over-infusion will happen if the coordinator receives bad data or has faulty software or application. While the latter can be addressed with proper design and software verification techniques, the former condition is a simple transformation from today's caregiver scenario: rather than a human receiving inaccurate data, the coordinator receives it. The action taken is largely the same. Hence, it is not sufficient to develop safe coordinator unless it also has secure communication.

- **All security attacks are manifest as a confused deputy attack.** We assume that the pump software itself is designed to meet certain safety goals. Thus, the pump can only violate patient safety goals if it receives invalid input from a caregiver or coordinator. Likewise, when the coordinator and the caregiver are both considered trusted, patients can only be harmed if the pump is mis-programmed based on inaccurate/delayed/partial inputs from the sensors and EHR.

- **Best safety practices may thwart some attacks.** The techniques used to prevent data entry errors for caregivers, such as drug libraries, barcode scanners, and flow sensors, also play a role in preventing security failures. However, these techniques may not be exhaustive nor sufficient to thwart all security attacks. In particular, each of these devices and their interconnects must be trustworthy; otherwise, an attacker can simply tamper with the information they provide to the coordinator and pump.

- **Only pervasive misinformation attacks can silence the interoperability coordinator.** The sensor inputs `ox_data` and `cap_data`, plus the pump output `pump_out` and possibly the EHR, must simultaneously be manipulated; otherwise, an alarm may be raised. Such an attack would require manipulation between the coordinator and pump, along with incorrect sensor data, to be effective.

- **Attacks from compromised entities in the interoperability are difficult to prevent.** If any of the three main types of entities in the interoperability setup, namely the sensors, the caregiver, and pump can be compromised, then the traditional information security solutions described for securing the inputs are rendered moot. One can use redundancy to attempt to detect events of compromise, but this requires at least one uncompromised IMD.

- **Security may be the proper subset of safety for IMDs**. When privacy is not considered (as is the case in our analysis), security may be a subset of safety. If we do consider privacy, then loss of privacy may not always lead to immediate safety problems for the patient. We do note that reconnaissance and eavesdropping are often precursors to more active attacks and that privacy may itself be an important security and safety goal.

6. RELATED WORK

Though some work has been done in developing frameworks for enabling interoperability between medical devices, little work has been done in exploring security issues for interoperable medical devices. King *et al.* [25] present an open-source Medical Device Coordination Framework (MDCF) for exploring solutions related to designing, implementing, verifying, and certifying systems of integrated medical devices. The framework supports a publish-subscribe architecture and uses a model-based programming environment for rapid development of IMD systems. The scope of this project has largely been on enabling interoperability and doing it safely in a certifiable manner [19]. A complimentary system called Network-Aware Supervisory System (NASS) has been proposed in [23] [36], which provides a development environment for safe medical device supervisory control in the presence of network failures. In [24], the authors have extended NASS to consider wireless networks. Both MDCF and NASS frameworks focus primarily on safe interoperation. Security has not been explored in either of the two frameworks.

In our previous work [34], the security of ICE architecture was examined assuming the devices were using a wireless channel to communicate. The analysis was a very high level and was not specific to any interoperability setting. In later work [32, 35], we developed high-level models for classifying the security attacks and their consequences on interoperable medical devices. These models again did not deal in the specifics of a particular interoperability setup and consequently cannot be used to aid in designing security-conscious interoperability architectures. That being said, models developed from these efforts are certainly complimentary to this effort and can be incorporated to extend this work.

7. CONCLUSIONS AND FUTURE WORK

Medical device interoperability is an increasingly prevalent example of how computing and information technology will revolutionize and streamline medical care. However, one aspect that has not been considered thus far is ensuring IMDs do not harm patients in the presence of malicious adversaries. This work outlines our effort in understanding the threats faced by IMDs. It is an important first step in eventually designing secure interoperability architectures. In this regard, we presented a detailed attack-graph-based analysis of threats on PCA interoperability under various levels of interoperability. Assuming a trusted coordinator, most of the attacks were discovered to be various forms of the confused deputy attack. We then described mitigation approaches possible for each of the possible attack classes. Many of the communication channel-oriented attacks can be mitigated using existing best-practices and available cryptographic solutions. However, entity-focused attacks based on physical compromise of the devices themselves are very difficult to protect against technologically. Our analysis shows that individual medical device safety does not equate to IMD safety despite having a trusted coordinator.

In the future, we plan to extend the analysis by removing the coordinator from the trusted computing base and analyze the potential for attacks on constituents of the coordinator, namely the supervisor and network controller, the logs and the alarm system. We also plan to expand on this effort to design an interoperability architecture and coordinator that can handle many of the security problems that the coordinator in the ICE architecture cannot handle. Overall, we want to understand the relationship between safety and security in IMDs and other such medical cyber-physical systems (MCPS), which, as of now, is not entirely clear.

8. REFERENCES

[1] D. Arney, S. Fischmeister, J. Goldman, I. Lee, and R. Trausmuth. Plug-and-Play for Medical Devices: Experiences from a Case Study. *Biomedical Instrument & Technology*, 43(4):313–317, July 2009.

[2] ASTM F29.21. Medical devices and medical systems — essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) — part 1: General requirements and conceptual model.

[3] A. Banerjee, S. K. S. Gupta, and K. K. Venkatasubramanian. Pees: Physiology-based end-to-end security for mhealth. In *In Proc. 4th Annual Wireless Health Conference*, Nov 2013.

[4] K. Banitsas, S. Tachakra, and R. S. H. Istepanian. Operational parameters of a medical wireless lan: security, range and interference issues. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, volume 3, pages 1889–1890 vol.3, 2002.

[5] T. Choen. Medical and information technologies converg. *IEEE Eng. Med. Biol. Mag*, 23(3):59–65, May–June 2004.

[6] M. Clarke, D. Bogia, K. Hassing, L. Steubesand, T. Chan, and D. Ayyagari. Developing a standard for personal health devices based on 11073. In *EMBS*, 2007.

[7] T. Denning, K. Fu, and T. Kohno. Absence makes the heart grow fonder: New directions for implantable medical device security. In *HotSec*, 2008.

[8] T. Denning, Y. Matsuoka, and T. Kohno. Neurosecurity: security and privacy for neural devices. *Neurosurgical Focus*, 27(1), 2009.

[9] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[10] FDA. Infusion pump improvement initiative. `http://www.fda.gov/downloads/MedicalDevices/ProductsandMedicalProcedures/GeneralHospitalDevicesandSupplies/InfusionPumps/UCM206189.pdf`, April.

[11] D. Foo Kune, K. Venkatasubramanian, E. Vasserman, I. Lee, and Y. Kim. Toward a safe integrated clinical environment: a communication security perspective. In *Proceedings of the 2012 ACM workshop on Medical communication systems*, MedCOMM '12, pages 7–12, 2012.

[12] K. Grifantini. Plug and Play Hospitals. `http://www.technologyreview.com/biomedicine/21052/`, July 2008.

[13] S. L. Grimes. Security: A new clinical engineering paradigm. *IEEE Eng. Med. Biol. Mag*, 23(4):80–82, July–August 2004.

[14] P. P. Gunn, A. M. Fremont, M. Bottrell, L. R. Shugarman, J. Galegher, and T. Bikson. The health

insurance portability and accountability act privacy rule: A practical guide for researchers. *Med. Care*, 42(4):321–327, April 2004.

[15] D. Halperin, T. Heydt-Benjamin, B. Ransford, S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *IEEE Security and Privacy*, 2008.

[16] S. Hammouda and Z. Trabelsi. An enhanced secure arp protocol and lan switch for preveting arp based attacks. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, IWCMC '09, pages 942–946, 2009.

[17] S. Hanna, R. Rolles, A. Molina-Markham, P. Poosankam, K. Fu, and D. Song. Take two software updates and see me in the morning: The case for software security evaluations of medical devices. In *USENIX conference on Health security and privacy*, 2011.

[18] N. Hardy. The confused deputy: (or why capabilities might have been invented). *ACM SIGOPS Operating Systems Review*, 22(4):36–38, 1988.

[19] J. Hatcliff, A. King, I. Lee, A. Macdonald, A. Fernando, M. Robkin, E. Vasserman, S. Weininger, and J. Goldman. Rationale and architecture principles for medical application platforms. In *IEEE/ACM Third International Conference on Cyber-Physical Systems (ICCPS)*, pages 3–12, 2012.

[20] Health level seven international. `http://www.hl7.org/`.

[21] Integrating the healthcare enterprise. `http://www.ihe.net/`.

[22] B. G. Kim, A. Ayoub, O. Sokolsky, I. Lee, P. Jones, Y. Zhang, and R. Jetley. Safety-assured development of the gpca infusion pump software. In *Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on*, pages 155–164, 2011.

[23] C. Kim, M. Sun, S. Mohan, H. Yun, L. Sha, and T. F. Abdelzaher. A framework for the safe interoperability of medical devices in the presence of network failures. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '10, pages 149–158, 2010.

[24] C. Kim, M. Sun, H. Yun, and L. Sha. A medical device safety supervision over wireless. In *Proceedings of the Reliable and Autonomous Computational Science*, RACS '10, pages 22–40, 2010.

[25] A. King, S. Procter, D. Andresen, J. Hatcliff, S. Warren, W. Spees, R. Jetley, P. Jones, and S. Weininger. An open test bed for medical device integration and coordination. In *Software Engineering - Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages 141–151, 2009.

[26] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. King, M. Mullen-Fortino, S. Park, A. Roederer, and K. Venkatasubramanian. Challenges and research directions in medical cyber physical systems. *Proceedings of the IEEE*, 100(1):75–90, 2012.

[27] Michael Wong. Physician-patient alliance for health and safety improving health and safety through innovation and awareness how often do errors with patient-controlled analgesia (PCA) occur? `http://ppahs.org/2011/10/31/how-often-do-errors-with-pca-occur/`.

[28] E. Morris, L. Levine, C. Meyers, D. Plakosh, and P. Place. Systems of systems interoperability. Technical report, Carnegie-Mellon University, April 2004.

[29] V. Pham and T. Aura. Security analysis of leap-of-faith protocols. In M. Rajarajan, F. Piper, H. Wang, and G. Kesidis, editors, *Security and Privacy in Communication Networks*, volume 96 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 337–355. Springer Berlin Heidelberg, 2012.

[30] N. L. Snee and K. A. McCormick. The case for integrating public health informatics networks. *IEEE Eng. Med. Biol. Mag*, 23(1):81–88, January–February 2004.

[31] A. Tolk, S. Diallo, and C. Turnitsa. Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering. *Journal of Systemics, Cybernetics and Informatics*, 5(5):65–74, 2007.

[32] E. Vasserman, K. Venkatasubramanian, O. Sokolsky, and I. Lee. Security and interoperable-medical-device systems, part 2: Failures, consequences, and classification. *IEEE Security & Privacy*, 10(6):70–73, 2012.

[33] K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta. Pska: Usable and secure key agreement scheme for body area networks. *Information Technology in Biomedicine, IEEE Transactions on*, 14(1):60–68, Jan 2010.

[34] K. Venkatasubramanian, S. Gupta, R. Jetley, and P. Jones. Interoperable medical devices. *Pulse, IEEE*, 1(2):16 –27, September–October 2010.

[35] K. Venkatasubramanian, E. Vasserman, O. Sokolsky, and I. Lee. Security and interoperable-medical-device systems, part 1. *IEEE Security & Privacy*, 10(5):61–63, 2012.

[36] P.-L. Wu, W. Kang, A. Al-Nayeem, L. Sha, R. B. Berlin, Jr., and J. M. Goldman. A low complexity coordination architecture for networked supervisory medical systems. In *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, ICCPS '13, pages 89–98, 2013.