CONTRIBUTED PAPER

Trustworthiness of Medical Devices and Body Area Networks

This paper surveys the threat landscape of medical embedded devices and the merits and shortcomings of existing defenses.

By Meng Zhang, Anand Raghunathan, *Fellow IEEE*, and Niraj K. Jha, *Fellow IEEE*

ABSTRACT | Implantable and wearable medical devices (IWMDs) are commonly used for diagnosing, monitoring, and treating various medical conditions. A general trend in these medical devices is toward increased functional complexity, software programmability, and connectivity to body area networks (BANs). However, as IWMDs become more ''intelligent,'' they also become less trustworthy—less reliable and more prone to attacks. Various shortcomings—hardware failures, software errors, wireless attacks, malware and software exploits, and side-channel attacks—could undermine the trustworthiness of IWMDs and BANs. While these concerns have been recognized for some time, recent demonstrations of security attacks on commercial products, e.g., pacemakers and insulin pumps, have elevated medical device security from the realm of theoretical possibility to an immediate concern. The trustworthiness of IWMDs must be addressed aggressively and proactively due to the potential for catastrophic consequences. Conventional fault tolerance and information security solutions, e.g., redundancy and cryptography, that have been employed in general-purpose and embedded computing systems cannot be applied to many IWMDs due to their extreme size and power constraints and unique usage models. While several recent efforts address defense of IWMDs against specific security attacks, a holistic strategy that considers all concerns and types of threats is required. This paper discusses trustworthiness concerns in IWMDs and BANs through a comprehensive identification and analysis of potential threats and, for each threat, provides a discussion of the merits and inadequacies of current solutions.

KEYWORDS | Body area networks (BANs); medical devices; personal healthcare systems; privacy; reliability; security

I. INTRODUCTION

Recent years have witnessed an explosion of activity in the development and use of implantable and wearable medical devices (IWMDs) for a variety of diagnostic, monitoring, and therapeutic applications. Advances in electronics promise to revolutionize the capabilities of IWMDs, leading to new generations of devices with increased functionality, programmability, and connectivity to body area networks (BANs). In addition, IWMDs are increasingly being connected to personal computers (PCs) and smartphones, and to web- or cloud-based medical data repositories, to provide patients with complete personal healthcare systems (PHSs). These advances, however, are shadowed by concerns about the trustworthiness—reliability and security—of the software and hardware deployed in such systems.

Due to their increasing functional complexity, ensuring the reliability of IWMDs is more challenging than ever. As devices become increasingly smaller in size, but more complex in both software and hardware, their design, testing, and eventual regulatory approval are becoming much more expensive for medical device manufacturers, both in terms of time and cost. The number of devices that have recently been recalled due to software and hardware defects is increasing at an alarming rate [1]. At the same time, the increasing programmability and

Manuscript received May 14, 2013; accepted April 23, 2014. Date of publication June 6, 2014; date of current version July 18, 2014. This work was supported by the National Science Foundation (NSF) under Grants CNS-0914787 and CNS-1219570.

M. Zhang and **N. K. Jha** are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: mengz@princeton.edu; jha@princeton.edu).

A. Raghunathan is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: raghunathan@purdue.edu).

Digital Object Identifier: 10.1109/JPROC.2014.2322103

network connectivity of IWMDs open them up to the possibility of malicious attacks. Recent demonstrations of successful attacks on medical devices, such as implantable cardioverter defibrillators (ICDs) [2] and insulin pumps [3], [4], have already placed them squarely in the focus of attackers.

Any concerns regarding trustworthiness in medical devices must be addressed aggressively and proactively due to the potential for catastrophic consequences. Unfortunately, IWMDs come with extreme size and power constraints, making it infeasible to simply borrow reliability or security solutions from the general-purpose computing arena. Therefore, this is an area that demands the immediate attention of the information security and embedded systems research communities, medical device manufacturers, and regulatory bodies.

This paper aims to provide a comprehensive analysis and categorization of the threats posed to the trustworthiness of IWMDs, and promising approaches to guard against them. In the following sections, we first provide some background on medical devices and BANs, and then analyze their reliability and security requirements. We next examine various types of threats that compromise the trustworthiness of these systems, and discuss suitable countermeasures against them.

II. PERSONAL HEALTHCARE SYSTEMS

This section provides relevant background on medical devices, and discusses how they are connected to BANs and various computing platforms to form PHSs.

A. Medical Devices

We start with an overview of medical devices.

The U.S. Food and Drug Administration (FDA) provides several classification standards for medical devices. Each device approved by the FDA is classified into one of the following 16 categories based on the medical specialty: anesthesiology; cardiovascular; clinical chemistry and clinical toxicology; dental; ear, nose, and throat; gastroenterology and urology; general and plastic surgery; general hospital and personal use; hematology and pathology; immunology and microbiology; neurology; obstetric and gynaecologic; opthalmic; orthopedic; physical medicine; and radiology [5].

Based on the potential risk for causing harm, the FDA provides another way of classifying medical devices [6]. Devices with a minimal risk, such as tongue depressors and handheld surgical instruments, belong to class I. Class I devices are subject to the least regulatory control: general controls. Medical devices with a higher risk, such as wheelchairs, surgical needles, and infusion pumps, are classified into class II. In addition to complying with general controls, class II devices are also subject to special controls, including special labeling, mandatory performance standards, postmarket surveillance, etc. Class III devices are more invasive and pose a much more significant risk, against which neither general nor special control is sufficient to assure safety and effectiveness. Such a device requires premarket approval, in addition to the general controls. Examples of such devices include replacement heart valves, cardiac pacemakers, and neurostimulators.

In this paper, we limit ourselves to IWMDs of two kinds: sensors that monitor the patient's ECG, temperature, blood glucose and oxygen levels, etc., and actuators that deliver therapies, such as cardiac pacing and drug injection. Sensors and actuators are often combined into a closed-loop system. For example, a pacemaker implanted into the human body includes a sensor that performs pacing measurements, and also an actuator that provides rate-adaptive bradycardia pacing support by analyzing the sensed data.

An actuator is usually equipped with a programmer to change configurations or issue commands wirelessly. For certain devices, such as pacemakers, programmers are available only in clinics or hospitals, and the patient must visit a qualified healthcare provider for device tuning. For other devices, such as insulin pumps, patient programmers are available that allow patients to adjust the devices to meet their needs at any time.

B. From IWMDs to BANs and PHSs

Fig. 1(a) and (b) presents some applications of IWMDs, including cardiac pacing, epileptic seizure detection, diabetes management through glucose monitoring and insulin delivery, etc. Fig. 1(c) presents a generic architecture for how IWMDs can be connected with each other via a BAN, as well as other computing platforms (mobile devices, PCs, and servers) to form a PHS. The system consists of four parts: medical sensors/actuators/programmers, a "hub" such as the patient's smartphone or a PC, a remote health server, and the doctor's smartphone or a PC.

The BAN facilitates communication among IWMDs and the hub. IWMDs may communicate with a patient's hub or with diagnostic equipment used by a healthcare provider, using short-distance communication technology, such as Medical Implant Communications Service [7], Bluetooth [8], ZigBee [9], or Ultrawideband Radios [10].

The hub is responsible for logging, compressing, and analyzing the raw health data recorded by the medical sensors. It serves as a bridge between the BAN and cellular service or an 802.11 wireless local area network (WLAN) to enable interaction between patients and remote healthcare providers. If any anomalies are detected in the readings from the sensors, the warnings and patient's location can be immediately transmitted to the healthcare provider.

Long-term data archival may be performed on a health server [11]–[13], which can be accessed by the patient or the healthcare provider. For example, a remote doctor can access the health server, query the patient's medical data, and provide a prescription or instructions, which may be displayed on the patient's hub.

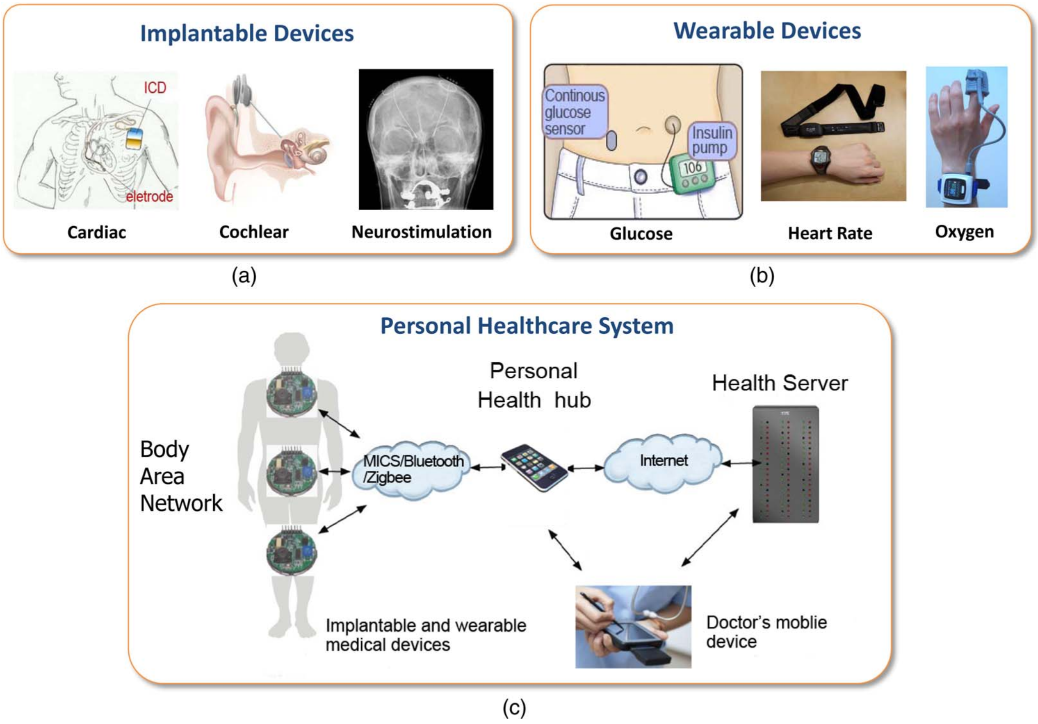


Fig. 1. *(a) Implantable and (b) wearable medical devices (adopted from http://www.wikipedia.org/), and (c) integration of IWMDs into a PHS.*

C. PHS Trustworthiness Requirements

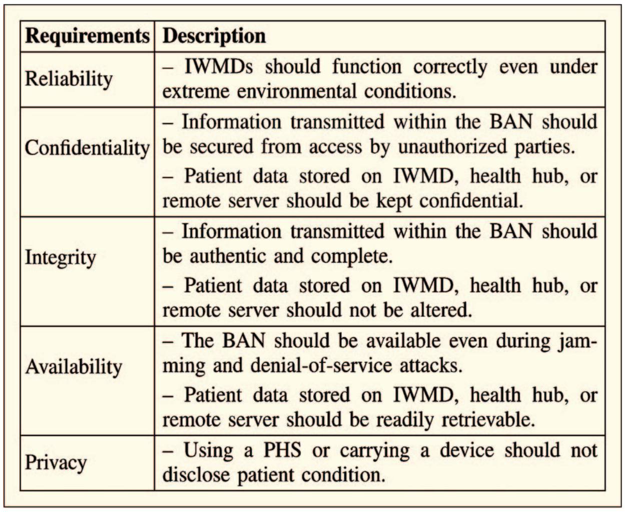
Unfortunately, any discussion on the potential of IWMDs and PHSs must be tempered by a sobering concern—trustworthiness. Studies suggest that security and safety are among patients' top concerns regarding IWMDs [14]. The sensitive nature of medical data and catastrophic consequences of safety failures or security breaches of PHSs make conventional reactive approaches unacceptable. Even when the consequences of a security attack are not life-threatening, the resulting negative perception and loss of trust may significantly slow down technology adoption. For example, the poor adoption of online health record services [11], [12] has been attributed in large part to security concerns [15].

Table 1 summarizes major PHS trustworthiness requirements.

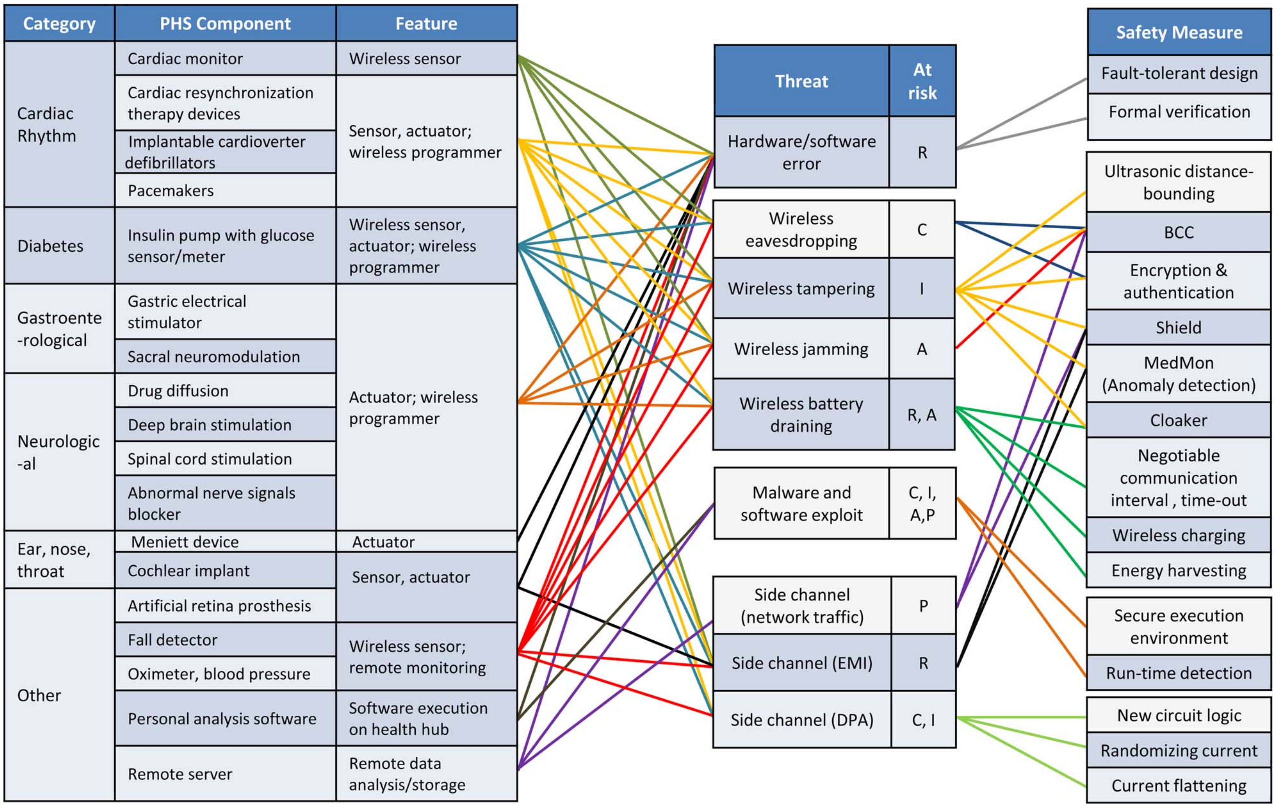
1) IWMD Reliability: Obviously, devices that perform life-sustaining functions must be reliable. If the device only provides monitoring capability and treatment is decided based on further assessment, the result of malfunction may not necessarily be harmful. For example, the wearable fall detector [16] for the elderly detects the occurrence of an unintentional fall and provides the location of the victim. A malfunctioning detector may send false alarms when the patient is safe and sound. In this case, the malfunction is apparent and the detector can be easily replaced. However, even for noncritical devices, malfunction can result in catastrophic consequences in indirect ways. For example, a visually impaired person aided by an artificial retina prosthesis [17] may suddenly lose vision while driving or crossing the street due to retina sensor failure.

2) BAN Security: Information leaked during wireless transmission can compromise BAN confidentiality and patient privacy. For example, the glucose monitoring and insulin delivery system discussed in [3] broadcasts the patient's diabetic condition when wirelessly transmitting

Table 1 PHS Trustworthiness Requirements



Requirements	Description
Reliability	– IWMDs should function correctly even under extreme environmental conditions.
Confidentiality	– Information transmitted within the BAN should be secured from access by unauthorized parties. – Patient data stored on IWMD, health hub, or remote server should be kept confidential.
Integrity	– Information transmitted within the BAN should be authentic and complete. – Patient data stored on IWMD, health hub, or remote server should not be altered.
Availability	– The BAN should be available even during jamming and denial-of-service attacks. – Patient data stored on IWMD, health hub, or remote server should be readily retrievable.
Privacy	– Using a PHS or carrying a device should not disclose patient condition.



Category	PHS Component	Feature
Cardiac Rhythm	Cardiac monitor	Wireless sensor
	Cardiac resynchronization therapy devices	Sensor, actuator; wireless programmer
	Implantable cardioverter defibrillators	
	Pacemakers	
Diabetes	Insulin pump with glucose sensor/meter	Wireless sensor, actuator; wireless programmer
Gastroenterological	Gastric electrical stimulator	Actuator; wireless programmer
	Sacral neuromodulation	
Neurological	Drug diffusion	
	Deep brain stimulation	
	Spinal cord stimulation	
	Abnormal nerve signals blocker	
Ear, nose, throat	Meniett device	Actuator
	Cochlear implant	Sensor, actuator
Other	Artificial retina prosthesis	
	Fall detector	Wireless sensor; remote monitoring
	Oximeter, blood pressure	
	Personal analysis software	Software execution on health hub
	Remote server	Remote data analysis/storage

Threat	At risk
Hardware/software error	R
Wireless eavesdropping	C
Wireless tampering	I
Wireless jamming	A
Wireless battery draining	R, A
Malware and software exploit	C, I, A,P
Side channel (network traffic)	P
Side channel (EMI)	R
Side channel (DPA)	C, I

Safety Measure
Fault-tolerant design
Formal verification
Ultrasonic distance-bounding
BCC
Encryption & authentication
Shield
MedMon (Anomaly detection)
Cloaker
Negotiable communication interval , time-out
Wireless charging
Energy harvesting
Secure execution environment
Run-time detection
New circuit logic
Randomizing current
Current flattening

Fig. 2. ***IWMDs, threats, and countermeasures. Reliability, confidentiality, integrity, availability, and privacy are referred to as R, C, I, A, and P, respectively.***

unencrypted data and commands. Even worse, an attacker can use the information (such as the device PIN) gained from eavesdropping on such communications to launch attacks and compromise BAN integrity. Depending on the device functionality, the consequence of an integrity compromise can be catastrophic, e.g., a cardiac pacemaker whose setting is changed by an attacker [2].

In addition to confidentiality and integrity requirements, the BAN and devices within it should be able to withstand attacks that are intended to cause unavailability, e.g., jamming attacks that flood the wireless channel, and denial-of-service attacks that drain the device battery [18].

3) Data Security: Just like BAN security, data security entails data confidentiality, integrity, and availability. Patient data should be kept safe when stored either on IWMDs, the hub, or the remote server. A compromised hub or a remote server can lead to a privacy breach. In addition, if treatment is decided based on the monitored data, corrupted data may lead to unsafe therapy, and missing/irretrievable data may impact the timeliness of treatment.

4) Privacy: In the context of this paper, privacy refers to hiding the patient condition from unauthorized parties. Privacy infringement can be a side effect of BAN/data confidentiality compromise, but may also occur in more subtle ways. It is important for IWMD manufacturers and healthcare providers to strive to protect patient privacy. For example, while a wheelchair user may not expect much privacy, it is understandable if a cancer patient wearing a tumor monitor [19] prefers to keep his condition a secret. If the attacker has knowledge of the wireless communication protocol used, the transmissions from an IWMD easily reveal its presence, and so does a response to a communication request sent by an attacker.

Fig. 2 shows examples of potential threats that can place a PHS at risk. Column 1 lists IWMDs and other PHS components. Column 2 shows the threats they are prone to. Column 3 includes possible defenses against the particular threats. We discuss these topics in detail next.

III. VULNERABILITY ASSESSMENT AND COUNTERMEASURES

In this section, we examine the various threats faced by IWMDs and other PHS components, all of which are capable of compromising a PHS and making its operation unsafe. These threats may arise from hardware/software errors or malicious attacks, which include radio attacks, side-channel attacks, malware attacks, and vulnerability exploits. We also discuss countermeasures against the threats.

Each of the following sections addresses one type of threat. Suitable countermeasures as well as their merits and inadequacies are also discussed. Some of these solutions are widely used, or have even been included in standards, whereas others are based on recent research.

A. Hardware Failures/Software Errors

In general, a hardware failure can be in electronic or nonelectronic components. Examples of nonelectronic hardware failure include contaminated syringes and broken wheelchairs. We focus on electronic hardware failure. Such a failure can be caused by undetected manufacturing defects, wear-and-tear faults due to electromigration, hot carrier degradation, dielectric breakdown, etc., as well as transient errors induced by a complex physical environment (e.g., due to noise, power disturbance, extreme temperature, vibration, electromagnetic interference, etc.). Studies have shown that electromagnetic interference may cause temporary or permanent malfunction in pacemakers and ICDs [20], [21].

Many IWMDs perform life-sustaining functions, such as cardiac pacing and defibrillation, and insulin delivery. They are also responsible for monitoring, recording, and storing private patient information, and making changes in response to doctors' orders. The critical nature of their functionality and the fact that they are in close contact with human organs leave little tolerance for hardware failure. A glitch on a cellphone may go unnoticed, whereas a glitch on a pacemaker or an ICD can be life-threatening. Besides cardiovascular and diabetic devices, neuromodulation devices that treat neurological conditions, such as Parkinson's disease and epilepsy [22], [23], are also examples that call for high standards for robustness and reliability. In a typical deep-brain stimulation system from Medtronic [24], several leads with electrodes are implanted in the brain and connected to a neurostimulator implanted near the clavicle. The amount of electrical stimulation can be noninvasively adjusted by a programmer. However, electromagnetic interference, if strong enough, could also change the parameter values of the neurostimulator, turn the neurostimulator off, or cause it to give the patient a jolt. A brain–machine interface (BMI) is expected to be at the center of the next generation of neuromodulation devices. It provides a direct communication pathway between the brain and an external device. BMI studies have shown promise for memory augmentation as well as perceptual and motor prostheses [25]. However, in addition to the technical challenges posed by interfacing of electronics with neurons, reliability concerns remain. Whether BMIs can reliably interact with the complex nervous system without being disrupted by environmental interferences needs further study.

Software reliability is just as important as hardware reliability. Many IWMDs are essentially embedded systems and have significant software content. Any safety and regulatory requirements for medical devices necessarily call for a rigorous software development process and skilled engineers in order to minimize software errors and protect public health. Unfortunately, designing bug-free software is difficult, especially in complex devices that might be used in unanticipated contexts. What is worse, software errors do not necessarily manifest themselves during the development and testing phase and may only result in errors after deployment.

More than a fourth of the recalls of defective medical devices during the first half of 2010 were likely caused by software defects [26]. Currently, there are no widely accepted techniques in use for the development or verification of software in medical devices [27]. The FDA is responsible for evaluating the risks of new devices and monitoring the safety and efficacy of those currently on the market. However, the FDA only assesses the development process of device software, not the integrity of the software itself. Unless a device that has already been surgically implanted repeatedly malfunctions or is recalled, the agency is unlikely to scrutinize the software operating on the device. Verification may often just depend on testing the device with crafted test cases, with little regard to the properties of the actual code. It has been argued that perhaps using open-source software is more secure and reliable for medical applications, as it enables continuous and broad peer review that identifies and eliminates software errors [26]. However, understandably, medical device manufacturers may be reluctant to adopt this approach.

Solutions: Next, we discuss two widely used techniques for enhancing hardware/software reliability.

1) Fault-Tolerant Design: Complex electronic circuits are used in life-critical healthcare systems, where reliability is of paramount importance. Though manufacturing-time test typically identifies a large number of circuit defects, exhaustive testing and attaining complete fault coverage may not be feasible. Through the concurrent detection, diagnosis, and correction of fault effects, fault-tolerant designs enable a system to continue operating properly in the event of faults in its components [28]. Fault tolerance can also be extended to cope with software errors caused by design inadequacies [29].

In general, fault tolerance requires some form of redundancy, either in time, hardware, or information. Hence, it incurs either performance degradation or hardware overhead. For example, triple modular redundancy (TMR) [30], a well-known fault tolerance scheme that employs three copies of a module and uses a majority voter to determine the final output, has more than three times the cost of the original circuit. Despite their high cost, fault-tolerant design techniques may be warranted in safety-critical medical devices.

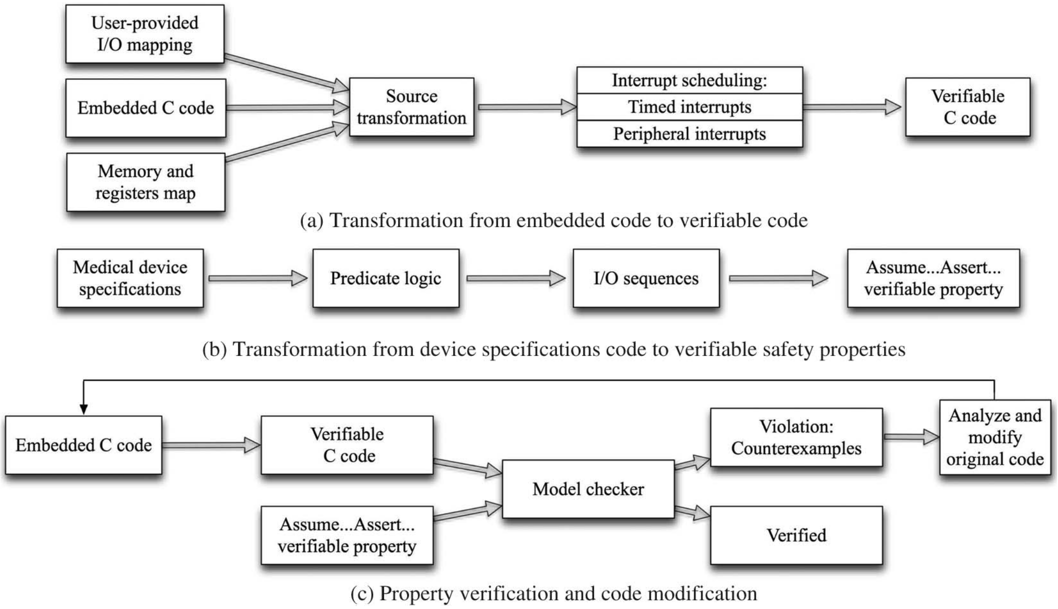

User-provided I/O mapping
Embedded C code
Memory and registers map
Source transformation
Interrupt scheduling:
Timed interrupts
Peripheral interrupts
Verifiable C code
(a) Transformation from embedded code to verifiable code
Medical device specifications
Predicate logic
I/O sequences
Assume...Assert... verifiable property
(b) Transformation from device specifications code to verifiable safety properties
Embedded C code
Verifiable C code
Assume...Assert... verifiable property
Model checker
Violation: Counterexamples
Analyze and modify original code
Verified
(c) Property verification and code modification

Fig. 3. ***Methodology for verifying safety properties of medical device software [34].***

2) Formal Verification: Formal methods have been suggested as a means to design and develop reliable medical systems [27], [31], [32]. Formal methods are mathematical techniques for the specification, development, and verification of software and hardware systems [33]. The specifications used in formal methods are well-formed statements in a mathematical logic, and formal verification consists of rigorous deductions in that logic. Therefore, formal methods provide a means to symbolically examine the entire state space of a system and establish that a correctness or safety property is true for all possible inputs.

Formal verification may be used to ensure that the software running on medical devices is free of vulnerabilities, such as buffer overflows. However, this is far from sufficient to ensure that the medical device would operate in a trustworthy manner. Two key challenges must be addressed in order to truly leverage the power of formal verification in the context of medical devices [34]. First, current software verification tools target specifications written in high-level programming languages, and are not suitable for the highly platform-specific and low-level programs that are written for medical devices. These programs interact with hardware peripherals, such as medical sensors and actuators, in addition to timers, ADCs, UARTs, etc. In addition, they often adopt a highly interrupt-driven software architecture. It is necessary to verify the operation of these programs with sufficient semantics of the hardware platform that they execute on, while avoiding the state–space explosion that results from excessive detail in modeling the hardware. Second, properties need to be verified at the interfaces of the medical devices with the real world. In other words, rather than merely verifying that a program is free of buffer overflows, it is equally, if not more so, important to verify whether any execution path in the pacemaker program leads to missing a cardiac pacing signal within a specified time window. A methodology to utilize formal verification for medical device software is shown in Fig. 3. The medical device software is first subject to a source-to-source transformation to address the aforementioned "semantic gap." Properties based on the functional specification of the medical device are then expressed in terms of input/output (I/O) interactions of the medical device with its environment, and translated into equivalent properties or assertions that must be satisfied by the medical device software. The transformed code and verifiable assertions are fed to a model checker, which verifies the code against the assertions and reports whether the code has been verified or a violation has been found.

B. Radio Attacks

A common IWMD design fallacy is relying on proprietary protocols for secrecy [35]. Since often no cryptographic protection is employed, wireless channels between devices and external controllers [e.g., the link between sensors and a smartphone in Fig. 1(c)], and between devices that communicate with each other, are highly prone to attacks.

A successful attack on an ICD is demonstrated in [2], which shows how the ICD design, which involves wireless communication with an external programmer, can be exploited by an attacker. By reverse-engineering the communication protocol, the attacker can launch radio attacks, with consequences ranging from disclosure of private data to alteration of device settings. In the worst case, the attacker can maliciously reconfigure the ICD to harm a patient by inaction (failure to deliver treatment when necessary) or by delivering an electrical stimulus when the heart is beating normally.

Using a similar approach, the study in [3] implements a successful attack on a glucose monitoring and insulin delivery system, exploiting both the wireless channel between the device and external controller, and the wireless channel between devices. The attacker first eavesdrops on the wireless packets sent from a remote control to an insulin pump, and reverse-engineers the communication protocol. The same eavesdropping attack is performed on a glucose meter that sends the glucose-level data to the insulin pump. The attacker discovers the PINs associated with the remote control and glucose meter. By mimicking the remote control, the attacker can configure the insulin pump to disable or change the intended therapy, stop the insulin injection, or inject a much higher dose than necessary. By mimicking the glucose meter, the attacker can send bogus data to the insulin pump, causing the patient to incorrectly adjust insulin delivery.

Another such man-in-the-middle attack is demonstrated on a Bluetooth-enabled pulse oximeter system in [36]. With the assumption that the PIN used in standard Bluetooth pairing is known, the attack shows that these wearable devices can be made to communicate with an unauthenticated intermediary equipped with a Bluetooth-enabled laptop.

Finally, with the knowledge of the communication protocol, denial-of-service attacks that aim to drain the battery of an implantable medical device (IMD) may be launched through the wireless channel. If the device responds to each incoming communication request from attackers, its battery may simply die and need to be surgically replaced. In addition, an attacker could also generate a large amount of noise to jam normal communication if he simply knows the approximate frequency of transmission.

Solutions: Next, we discuss several methods to detect, defend against, or mediate radio attacks. They are classified into four categories: close-range communication, cryptography, external device, and battery-constraint mitigation.

1) Close-Range Communication: Limiting the communication range is a simple and intuitive way of limiting radio attacks. A radio-frequency identification (RFID)-based channel between medical devices and external controllers is often proposed in this context [37]. However, an attacker with a strong enough transmitter and a high-gain antenna can attack the wireless channel even if the channel is only for RFID-based communication. For an RFID channel, the attacker can access the IWMD from up to ten meters away [38], [39]. A better alternative may be near-field communication (NFC), an extension of RFID, which is gaining increasing attention, especially due to its integration on mobile phones [40]. The typical working distance for NFC is up to 20 cm. However, there is no guarantee that an attacker with a high-gain antenna cannot read the signal from outside the intended range, e.g., from 1 m away [41].

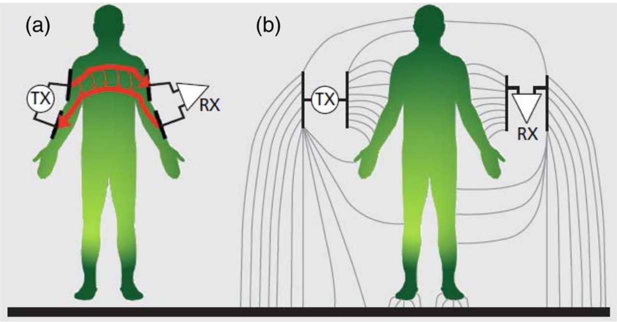


Fig. 4. ***Two coupling mechanisms for body-coupled communication [42].***

Another technology that can help limit the communication range is body-coupled communication (BCC). In contrast to conventional wireless communication, BCC uses the human body as the transmission medium. The communication range is limited to the proximity of the human body. Fig. 4 illustrates two coupling mechanisms for BCC. In Fig. 4(a), electrodes are directly attached to the human body for transmitting electrical signals. In Fig. 4(b), the human body acts as a floating conductor, whose electric potential changes with the electric field generated by the transmitter, which is detected by the receiver [42]. Experimental results presented in [3] show a promising attenuation in signal strength measured from some distance when comparing the BCC channel signal to the air channel signal. However, these radios work at low frequencies (ranging from 10 kHz to 10 MHz) and can only achieve very low data rates.

In addition to communications that are designed to be inherently short range, measures can be taken to enforce close-range communication. An access control scheme based on ultrasonic distance bounding is introduced in [43]. In this scheme, an IWMD grants access to its resources to only those devices that are close enough. Shielding is another way of enforcing close-range communication. A metal shield that restrains wireless signals from traveling beyond it can effectively eliminate radio eavesdropping attackers at a distance. However, limiting the communication range is only effective against radio attacks launched from beyond a certain distance. It is quite possible that an attacker can approach within a small distance of the patient and even make physical contact without raising suspicion (e.g., in a crowded subway station). Therefore, close-range communication schemes cannot defend against all close-range attacks.

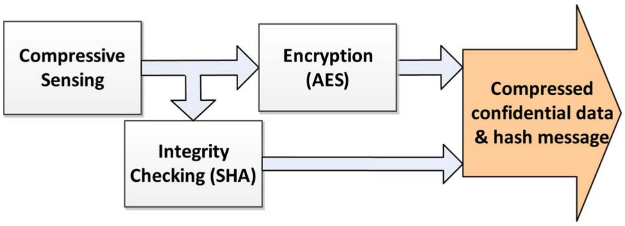


Fig. 5. *Encompression based on compressive sensing [45].*

2) *Cryptography:* Cryptography is the best approach for securing the wireless communication channel and preventing unauthorized access. The high energy and implementation costs of asymmetric cryptography preclude its use for encrypting medical data in IWMDs, which leaves symmetric encryption as the only practical option. However, the use of symmetric ciphers may still greatly increase the energy consumption and thus shorten device battery time. To mitigate this problem, compression techniques can be used before encryption to reduce this added workload as well as transmission cost. Compressive sensing [44] is particularly well suited since the compression can be realized with a very low computational and energy footprint. An evaluation of encompression (compressive sensing + encryption + integrity checking) [45] shows that an energy reduction of up to 78% can be achieved using encompression versus traditional encryption and integrity checking, with a reasonable compression ratio of 6–10×. An overview of the encompression scheme is shown in Fig. 5. The advanced encryption standard (AES) [46] is used as the symmetric cipher and the secure hash algorithm (SHA) [47] is used for integrity checking. Note that the hash algorithm must be applied on the original data (the plaintext), because then imposters cannot generate encrypted data whose plaintext matches the hash output, without knowing the AES secret key. If it is applied on the encrypted data (the ciphertext), imposters can hash and send random spurious data if they have knowledge of the hash function used.

Cryptographic methods are even more attractive when the secret keys shared by IWMDs and the hub can be renewed periodically, as in 802.11 WiFi [48]. Fixed preconfigured keys, which are discussed in Section III-D, are prone to attacks. Furthermore, the secret keys should be updated automatically, since many users, such as the elderly, are unable or unwilling to configure keys of sufficient strength, or update them frequently. Ideally, shared keys should be generated with high agreement (low mismatch rate between the two communication parties), high randomness, at a fast rate, and with a minimum computational/energy overhead. Unfortunately, as always, tradeoffs must be made among these conflicting goals. Refreshing at a low rate (e.g., 1 b/s) can lower disagreement, computational/energy overhead, and improve randomness, and may just suffice for low-data-rate IWMDs. Works described in [49]–[51] focus on extracting shared secret keys over unsecured wireless channel using the directional symmetry of wireless links. The most commonly used metric in these works is the received signal strength indicator (RSSI), a measure of signal power in logarithmic units. However, generation of randomness relies on relative movement between sensors and the hub, or the environment being dynamic [51]. In the case where both IWMDs and the hub are mounted on the body at fixed positions, the measured RSSI traces may not provide enough secrecy. In addition, rapid fluctuations in signal strength due to blockage by clothes or sudden movement may cause asymmetry in signals received at the two ends of the communication, which may result in key disagreement and require reconciliation.

Authentication schemes using properties of human bodies as alternatives to input of passwords have been proposed. IMDGardian [52] introduces an alternative cryptographic scheme for implantable cardiac devices that utilizes the patient's electrocardiography signals for key extraction. The identification system presented in [53] successfully recognizes people with an accuracy of 90% by measuring their bioimpedance to alternating current of different frequencies.

Unfortunately, conventional cryptographic methods are not directly applicable to IWMDs, whose unique usage models may require key distribution to legitimate parties outside the BAN. For example, encryption prevents medical professionals from accessing the patient's health data in emergency situations. As a possible solution, a universal key may be preloaded in devices of the same model that the ambulance staff can request from the manufacturer or patient's doctor in emergencies. However, this scheme is inherently unsafe as attackers can discover the secret key of a particular model through side-channel attacks or by hacking into the doctor's computer. Another straightforward key-distribution solution is to ask patients to carry cards or bracelets imprinted with the secret keys of their devices. To prevent the imprints from being lost or damaged, the keys could be printed into the patient's skin using ultraviolet-ink micropigmentation [54]. These "tattoos" only become visible under ultraviolet light, which is how the ambulance staff can find the keys and access the devices. To some extent, this approach is secure against close-range attacks, since although the attacker may be in close proximity, it is unlikely that the attacker can lift up the patient's sleeves while shining ultraviolet light without raising suspicion.

3) *External Device:* To preserve IMD battery power, verification of incoming requests can be offloaded to a trusted external device, which, unlike IMDs, can be easily recharged. A wearable device, called Communication Cloaker, is described in [55]. The Cloaker mediates communications between the IMD and preauthorized parties

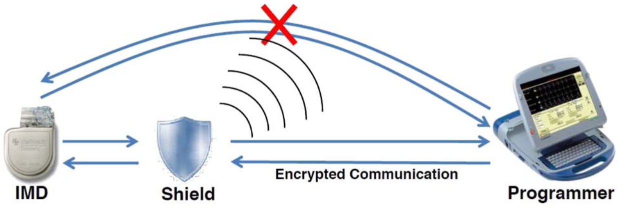


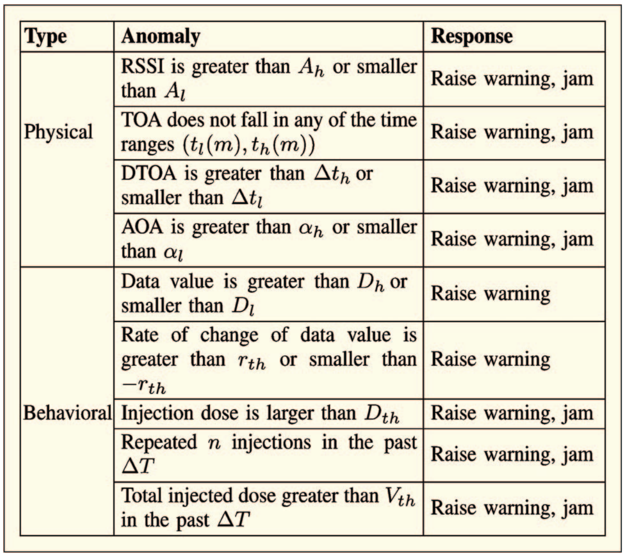
Fig. 6. ***External device that relays communications between the IMD and the programmer [56].***

and causes the IMD to ignore incoming communications from all unauthorized programmers. If the Cloaker is missing or broken, the IMD accepts and responds to all incoming communications. Therefore, in emergency situations, the medical staff can remove the Cloaker in order to access the IMD. Since the burden of computation is offloaded to the external device, this approach can protect the IMD against battery-draining attacks.

Another external device, a personal base station called the "shield," is described in [56]. It is illustrated in Fig. 6. The shield works as a relay between the IMD and the external programmer. It is designed to receive and jam the IMD messages at the same time, so that others cannot decode them. It then encrypts the IMD message and sends it to the legitimate programmer. The shield also protects the IMD from unauthorized incoming commands by jamming all the messages sent directly to the IMD. All commands must be encrypted and sent to the shield first, which then relays legitimate commands to the IMD. Therefore, the shield does not require any change in commercial IMDs, but requires changes in all programmers. Since the messages from the IMD are jammed and the communication between the programmer and the shield is encrypted, the confidentiality of IMD messages is protected. However, when the shield sends programmer's commands to the IMD, confidentiality is not warranted.

A medical security monitor (MedMon), proposed in [57], snoops on all the radio-frequency wireless communications to/from medical devices and uses anomaly detection to identify potentially malicious transactions. Anomalies are detected through physical characteristics of the transmitted signal, such as the RSSI, the time of arrival (TOA), the differential time of arrival (DTOA), and the angle of arrival (AOA), or behavioral characteristics embedded in the underlying information. Upon detection of a potential malicious transaction, MedMon takes appropriate response actions, which could range from passive (notifying the user) to active (jamming the packets so that they do not reach the medical device). Table 2 shows examples of security policies that could be used. These policies may be different for different IWMDs. The parameters mentioned in the table can be set to predefined values or else tailored to the patient's condition and environment. Values of parameters associated with physical anomalies, e.g., A_h and Δt_h, can be generated automatically at the end of a setup period. Values of parameters associated with behavioral anomalies, e.g., D_{th} and r_{th}, can be selected based on the advice of the doctor. By acting like a firewall, the monitor protects the BAN against integrity attacks, which are arguably the most dangerous type of attacks. It does not protect BAN confidentiality and patient privacy against eavesdropping, nor does it protect BAN availability against jamming. However, it does have the ability to provide protection for IWMD availability against battery-draining attacks, as transmissions that are too frequent may be seen as malicious and jammed. A key benefit of MedMon is that it is applicable to both existing IWMDs and programmers with no hardware or software modifications needed for them. Consequently, it leads to zero power overheads on these devices.

Table 2 Examples of Security Policies [57]



Type	Anomaly	Response
Physical	RSSI is greater than A_h or smaller than A_l	Raise warning, jam
	TOA does not fall in any of the time ranges $(t_l(m), t_h(m))$	Raise warning, jam
	DTOA is greater than Δt_h or smaller than Δt_l	Raise warning, jam
	AOA is greater than α_h or smaller than α_l	Raise warning, jam
Behavioral	Data value is greater than D_h or smaller than D_l	Raise warning
	Rate of change of data value is greater than r_{th} or smaller than $-r_{th}$	Raise warning
	Injection dose is larger than D_{th}	Raise warning, jam
	Repeated n injections in the past ΔT	Raise warning, jam
	Total injected dose greater than V_{th} in the past ΔT	Raise warning, jam

4) Battery-Constraint Mitigation: Compared to wearable devices whose battery can be readily recharged or replaced, battery-draining attacks pose much greater threat to IMDs, such as EEG implants and pacemakers, since replacing the battery usually implies surgery. Zero-power defenses (security at no cost to the battery) have been proposed for ICDs, in which the induced RF energy is harvested for notification, authentication, and key exchange [2]. In addition, efforts are being undertaken to design BAN protocols to mitigate this problem. For example, the IEEE 802.15.6 BAN standard allows a node and a hub to negotiate their communication intervals by encoding them in authenticated messages. The node thus will not wake up to receive any messages outside the negotiated time intervals [58].

Relaxing the battery constraints would be the best defense against battery-draining attacks. One solution is to make the implant wirelessly rechargeable [59]. Another is to harness kinetic energy from the human body [60].

However, as intriguing as these new techniques may sound, they are still in the research phase and must go through rigorous testing and examination to ensure trustworthiness before commercial use.

C. Malware and Vulnerability Exploits

Various forms of malware, including viruses, worms, Trojans, keyloggers, botnets, and rootkits, have emerged and keep evolving and adapting to new platforms. Smartphone platforms, such as Android and iOS, have been breached by mobile malware [61], [62]. With the increasing flexibility and connectivity of PHS platforms, it is just a matter of time before the first appearance of malware that targets PHS platforms. For example, Intel Health Guide [13] is a chronic care product that delivers personalized health monitoring at home. Patients can use the system to measure their own vital signs and, through the Internet, upload the results to a remote server, where healthcare professionals can assess the patient's health condition. A virus that infects such a system can delete or forge health data.

Furthermore, since software is inherently complex, abstract, and intangible, software vulnerabilities are inevitable and difficult to detect. Software vulnerabilities differ from software errors in that they are not logical errors, but unsafe code segments that can be exploited by attackers. For example, use of especially crafted inputs, which trigger buffer overflows and redirect the program to execute malicious code, is called a buffer overflow attack. The corrupted memory could originally be holding an address to an instruction, which the program should be redirected to. After corruption of the address, the program may be redirected to a false address and start executing random code. With some knowledge of system software, attackers can exploit the buffer overflow vulnerabilities as well as other software vulnerabilities to steal private information, tamper with medical data, and even change device settings.

While BANs are subject to unique threat models and attacks, as described in previous sections, software attacks will continue to be a commonly utilized approach for compromising their security, due to the relative ease and low cost of launching such attacks. In this context, the "weakest link" of a BAN, i.e., the component that exposes the largest attack surface and is the most accessible to software attacks, is the health hub, which executes the medical applications (for logging of health data, display of data to the user, and communication with remote medical professionals and health information services). As reflected by the rapid proliferation of "application" marketplaces for mobile devices, users are likely to use their smartphones to execute untrusted and potentially vulnerable applications as well. In the extreme case, the operating system (OS) on the hub may itself be compromised, making it trivial to subvert the medical applications that execute under its full control. Thus, it becomes essential to provide a secure execution environment for the medical applications in the face of other untrusted applications and also an untrusted OS.

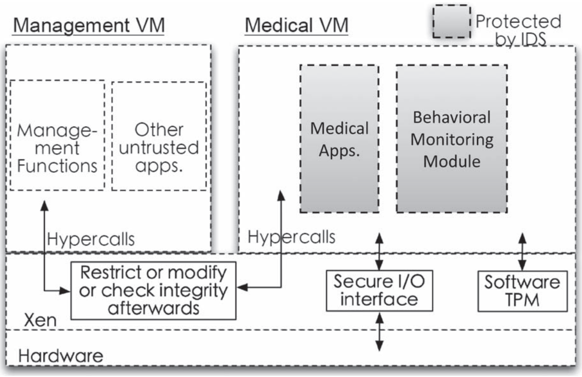


Fig. 7. *Secure execution environment for medical applications.*

Solutions: Two sets of techniques, secure execution environment and runtime monitoring, can be used to defend medical software/data against malware, vulnerability attacks, and malicious OS.

1) Secure Execution Environment: While it may not be feasible to secure all applications from a compromised OS, it is possible to achieve a secure execution environment that provides isolation for selected, security-critical applications. The isolation may be based on physical separation (e.g., IBM's secure coprocessor [63]) or logical separation, in which both the sensitive and untrusted codes are run on the same processor, but are isolated either using an additional layer of software, or through additional hardware support, such as ARM TrustZone [64].

A secure execution environment based on logical separation for medical applications is illustrated in Fig. 7. It is based on two key technologies: secure virtualization and trusted computing. For PHSs, virtualization is a promising technology that can be utilized to enhance security by providing isolated execution environments for different applications that require different levels of security. The medical applications are the most security-critical components. As shown in Fig. 7, they can be protected in a separate virtual machine (VM), which we refer to as the medical VM. The medical VM is a restricted environment in which only medical applications and the supporting software libraries are executed, isolated from the other applications running on the system. Trusted computing [65] is a set of standards that is widely gaining popularity in general-purpose computing systems. Trusted computing requires a "root of trust" in the system for tamper-proof storage and attestation, which is typically realized by adding a separate tamper-proof hardware component called the trusted platform module (TPM) to the system. In size-constrained and resource-constrained platforms, such as

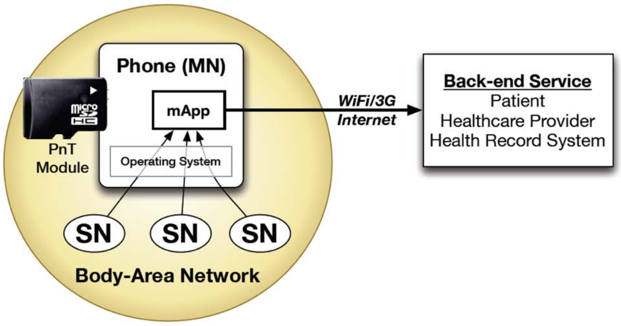


Fig. 8. ***Data separation by a plug-in smart card. The smartphone OS only has an encrypted view of medical data [67].***

smartphones, it is currently not common to see hardware TPMs. In such cases, the use of a software TPM based on software emulation of TPM functions within an isolated execution environment has been demonstrated [66].

In addition to logical separation, the goal of data confidentiality and integrity can also be achieved by physically separated and secured data storage on the hub (smartphone). For example, Plug-n-Trust [67] is a plug-in smart card that provides a trusted computing environment and keeps medical data safe. The principle of Plug-n-Trust is as follows. Assuming the data sent by the medical sensors are encrypted, they remain encrypted while stored on the hub, and are only decrypted within the smart card. Application programming interfaces (APIs) are provided by the card to allow data modification by medical applications. This model is illustrated in Fig. 8.

A more aggressive approach is to completely separate health-related applications from untrusted applications/ OS by making the hub an independent device. One such wrist-worn device, called Amulet (Fig. 9), is proposed in [68]. Amulet is dedicated to communications with IWMDs. It occasionally communicates with the smartphone in order to connect with health servers. It can also authenticate its wearer and determine which set of sensors are on the body by using techniques introduced in [69] and [70]. In addition to physical separation from potential software attacks, another strong argument in favor of a dedicated hub is that IWMDs must be able to operate continuously and securely without relying on smartphones or other nonwearable personal computing devices, which can easily be lost, stolen, or run out of power.

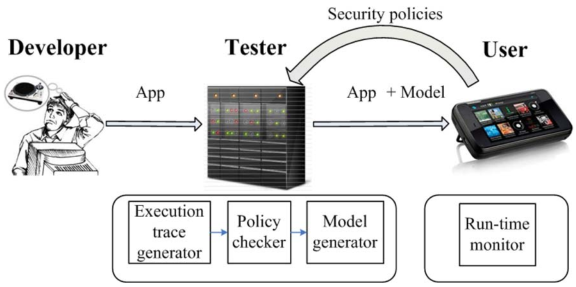


Fig. 10. ***Software runtime monitor. Applications are first tested in VMs and monitored when actually used.***

2) Runtime Monitoring: Isolating medical applications from other software does not protect against vulnerabilities in the medical applications themselves, which are commonly introduced into software as artifacts of the software development process. Intrusion detection techniques based on dynamic binary instrumentation have been extensively investigated [71]. As shown in Fig. 10, the application is first tested by running against a large input set (manually crafted or automatically generated) in a virtualized environment. If it passes the test, its behavioral models are generated, which can be seen as a database of good behaviors. The user may also request the publisher to test the application against user-defined policies that are of most interest to him/her. Runtime monitoring at the user end restricts the application's behavior to within the database of good behaviors. Any deviation is detected as an anomaly. As much of the workload is shifted from the user

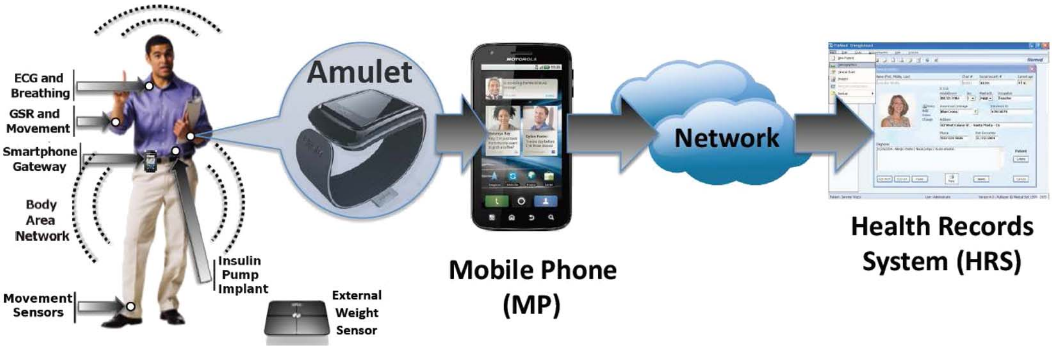


Fig. 9. ***Dedicated wrist-worn device as the hub [68].***

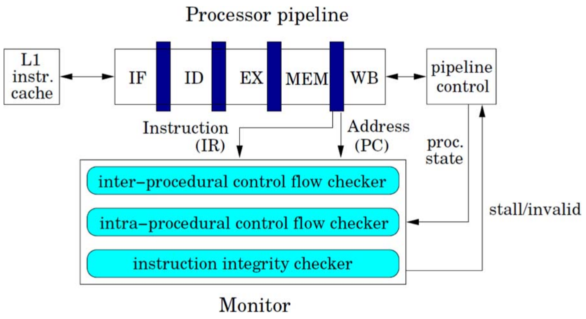


Fig. 11. ***Hardware-assisted runtime monitor that monitors cycle-by-cycle trace of the executing instructions and their program addresses [72].***

to the testing end, the performance penalty is greatly reduced compared to rigorous runtime checking.

Even though the monitoring work is minimized, the delay overhead for such fine-grained monitoring can still be prohibitive, especially for applications with intensive user interactions. To overcome this problem, a hardware-assisted runtime monitor has been proposed for secure embedded processing [72]. Fig. 11 shows the conceptual block diagram of such a hardware monitor. The embedded processor is depicted as an in-order five-stage pipeline. It is augmented with a hardware monitor that observes the processor's dynamic execution trace, checks whether the execution trace falls within the allowed program behavior, and flags any deviations from the expected behavior to trigger appropriate response mechanisms. Program behaviors can be represented at different levels of granularity, namely, interprocedural control flow, intraprocedural control flow, and instruction stream integrity. When the monitor detects a violation of permissible program behavior, it asserts the invalid signal. In rare cases when the monitor is unable to keep pace with processor execution, it asserts the stall signal.

D. Side-Channel Attacks

Side-channel attacks exploit information leaked through physical channels, such as power consumption, execution time, electromagnetic emission, etc. [73]–[75]. They can be used against medical devices and PHSs for privacy invasion, as discussed in Section II-C4. For example, the Intel Health Guide system [13] is equipped with integrated cameras, allowing online health sessions and video consultations through the Internet. However, the network traffic flow may leak patients' private information. The schedule of health sessions and video calls, for example, could be deduced from monitoring the network traffic flow. One could also infer a change in the patient's health condition, if the lengths and frequencies of health sessions and video calls suddenly increase.

A more dangerous type of side-channel attack exploits electromagnetic interference (EMI), which can affect the circuit by inducing voltage on conductors. Analog sensors in IWMDs are particularly susceptible. It has been shown that EMI can inhibit pacing and induce defibrillation shocks on implantable cardiac devices at a close distance [76].

Another form of side-channel attack is the differential power analysis (DPA) attack. A DPA attack can extract secret keys from extremely noisy signals and is very difficult to guard against. It employs statistical analysis of measured power consumption traces, which are correlated with the data handled by the physical device [73]. Although no known DPA attacks on any medical device have been reported, it is not hard to construct a scenario where DPA breaks cryptographic protection on IWMDs. Suppose a heart-rate monitor uses a symmetric block cipher (such as AES) with a built-in secret key to encrypt the measured heart rates before sending them to the hub. If an attacker gains access to the heart-rate monitor, the secret key can easily become a vulnerable target of DPA and extracted by feeding it with various data, measuring the corresponding current consumption, and analyzing the difference in measured current traces. Successful recovery of the secret key would then compromise confidentiality. Even worse, if a common default key is used for all shipped units of the same model, the attacker could publicize the revealed secret key and thus make the cryptographic protection ineffectual.

Solutions: Next, we introduce some of the proposed countermeasures against EMI and DPA attacks. Other types of side-channel attacks (e.g., cache attacks) and their countermeasures are omitted because we believe they are less applicable to the PHS model.

1) Countermeasures Against EMI: Shielding and filtering are commonly used defenses against EMI. In addition, cardiac defense mechanisms may take advantage of the physical proximity to the human body and detect suspicious sensor inputs by checking whether pacing pulses are consistent with the refractory period of cardiac tissue [76]. This method falls under the category of anomaly detection.

2) Countermeasures Against DPA: Software solutions against DPA, such as key masking [77] , which attempts to randomize the secret key prior to each execution of the scalar multiplication under analysis, incur too much energy overhead. Assists from hardware design are usually proposed.

As the reason for the vulnerability of classical CMOS logic circuits to DPA attacks lies in the imbalance of charging and discharging behavior between 0-to-1 and 1-to-0 transitions, novel logic styles with data-independent power consumption have been proposed as circuit-level solutions to reduce the dependence of power dissipation on input patterns [78], [79].

Other system-level countermeasures either try to suppress the differential signal used in the DPA attacks or randomize the power profile. For example, in [80], a current flattening circuit is introduced. In [81], an additional circuit serving as a bandpass filter is added to the cryptosystem to suppress information leakage through the current supply pin. In [82], an internally generated random mask based on ring oscillators is used to dynamically change the power consumption. In [83], a dynamic voltage and frequency switching approach is adopted in which both the voltage and the clock frequency can be dynamically selected by a processor. Random delays are inserted in the datapath in [84] and FinFET back-gate biasing is introduced in [85] to randomize the power profile.

Unfortunately, in most of the aforementioned methods, DPA resistance still comes at the expense of large area and power overheads, which are not compatible with resource-constrained IWMDs. Low-cost DPA-resistant design is still an open problem.

IV. CONCLUSION

A general trend in IWMDs is toward increased functional complexity, software programmability, and wireless network connectivity. An undesirable, yet inevitable, side effect of these trends is that IWMDs and BANs are increasingly vulnerable to security attacks. Trustworthiness concerns may become a hindrance to further commercialization of IWMDs and BANs. We analyzed various aspects of threats faced by them and discussed suitable solutions for each threat. Given the critical functions IWMDs perform, these issues should be addressed aggressively and proactively by the manufacturers before market deployment. ■

Acknowledgment

The authors would like to thank Prof. S. Sur-Kolay for her valuable feedback on the paper.

REFERENCES

[1] Stericycle Expert Solutions, *FDA Recall Trends for Food, Drugs, Devices*, 2011. [Online]. Available: http://www.expertrecall.com

[2] D. Halperin *et al.*, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Proc. IEEE Symp. Security Privacy*, May 2008, pp. 129–142.

[3] C. Li, A. Raghunathan, and N. K. Jha, "Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system," in *Proc. IEEE Int. Conf. e-Health Netw. Appl. Services*, Jun. 2011.

[4] J. Radcliffe, "Hacking medical devices for fun and insulin: Breaking the human SCADA system," in *Proc. Black Hat Technical Security Conf.*, Jul./Aug. 2011.

[5] U.S. Food and Drug Administration, *Device Classification*, 2009. [Online]. Available: http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/Overview/ClassifyYourDevice

[6] U.S. Food and Drug Administration, *General and Special Controls*, 2009. [Online]. Available: http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/Overview/GeneralandSpecialControls

[7] H. S. Savci, A. Sula, Z. Wang, N. S. Dogan, and E. Arvas, "MICS transceivers: Regulatory standards and applications," in *Proc. IEEE SoutheastCon*, Apr. 2005, pp. 179–182.

[8] Bluetooth. [Online]. Available: http://www.bluetooth.org

[9] Zigbee alliance. [Online]. Available: http://www.zigbee.org

[10] D. Porcino and W. Hirt, "Ultra-wideband radio technology: Potential and challenges ahead," *IEEE Commun. Mag.*, vol. 41, no. 7, pp. 66–74, Jul. 2003.

[11] Google Health Service. [Online]. Available: http://www.google.com/health

[12] Microsoft HealthVault. [Online]. Available: http://www.healthvault.com

[13] Intel Health Guide, 2011. [Online]. Available: http://www.careinnovations.com/products/healthguide

[14] T. Denning *et al.*, "Human values and security for wireless implantable medical devices," in *Proc. Int. Conf. Human Factors Comput. Syst.*, Apr. 2010.

[15] M. Kolbasuk McGee, "5 reasons why Google Health failed," *InformationWeek*, Jun. 29, 2011. [Online]. Available: http://www.informationweek.com/news/healthcare/EMR/231000697

[16] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," in *Proc. IEEE Int. Conf. Eng. Med. Biol. Soc.*, Jan. 2005, pp. 3551–3554.

[17] L. Schwiebert *et al.*, "A biomedical smart sensor for the visually impaired," in *Proc. IEEE Sensors*, 2002, vol. 1, pp. 693–698.

[18] T. Martin, M. Hsiao, D. Ha, and J. Krishnaswami, "Denial-of-service attacks on battery-powered mobile computers," in *Proc. IEEE Conf. Perv. Comput. Commun.*, Mar. 2004, pp. 309–318.

[19] K. D. Daniel *et al.*, "Implantable diagnostic device for cancer monitoring," *Biosens. Bioelectron.*, vol. 24, no. 11, pp. 3252–3257, Jul. 2009.

[20] D. L. Hayes *et al.*, "Interference with cardiac pacemakers by cellular telephones," *New England J. Med.*, vol. 336, no. 21, pp. 1473–1479, May 1997.

[21] C. Jilek *et al.*, "Safety of implantable pacemakers and cardioverter defibrillators in the magnetic field of a novel remote magnetic navigation system," *J. Cardiovasc. Electrophysiol.*, vol. 21, no. 10, pp. 1136–1141, Oct. 2010.

[22] J. M. Bronstein *et al.*, "Deep brain stimulation for Parkinson disease: An expert consensus and review of key issues," *Arch. Neurol.*, vol. 68, no. 2, pp. 165–171, Feb. 2011.

[23] W. H. Theodore and R. S. Fisher, "Brain stimulation for epilepsy," *Lancet Neurol.*, vol. 3, no. 2, pp. 111–118, Feb. 2004.

[24] Medtronic, 2011. [Online]. Available: http://www.medtronic.eu/our-therapies

[25] A. Csavoy, G. Molnar, and T. Denison, "Creating support circuits for the nervous system: Considerations for brain-machine interfacing," in *Proc. Int. Symp. VLSI Circuits*, Jun. 2009, pp. 4–7.

[26] K. Sandler, L. Ohrstrom, L. Moy, and R. McVay, *Killed by Code: Software Transparency in Implantable Medical Devices*, 2010. [Online]. Available: http://www.softwarefreedom.org

[27] D. Arney, R. Jetley, P. Jones, I. Lee, and O. Sokolsky, "Formal methods based development of a PCA infusion pump reference model: Generic infusion pump (GIP) project," in *Proc. Joint Workshop High Confidence Med. Devices Softw. Syst. Med. Device Plug-and-Play Interoper.*, Jun. 2007, pp. 23–33.

[28] W. R. Moore, "A review of fault-tolerant techniques for the enhancement of integrated circuit yield," *Proc. IEEE*, vol. 74, no. 5, pp. 684–698, May 1986.

[29] B. Randell, "System structure for software fault tolerance," *SIGPLAN Not.*, vol. 10, pp. 437–449, Apr. 1975.

[30] R. E. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM J. Res. Dev.*, vol. 6, pp. 200–209, Apr. 1962.

[31] R. Jetley, S. P. Iyer, P. L. Jones, and W. Spees, "A formal approach to pre-market review for medical device software," in *Proc. Int. Conf. Comput. Softw. Appl.*, Sep. 2006, pp. 169–177.

[32] L. Cordeiro, B. Fischer, H. Chen, and J. Marques-Silva, "Semiformal verification of embedded software in medical devices considering stringent hardware constraints," in *Proc. IEEE Int. Conf. Embedded Softw. Syst.*, May 2009, pp. 396–403.

[33] C. M. Holloway, "Why engineers should consider formal methods," in *Proc. IEEE Digit. Avionics Syst. Conf.*, Oct. 1997, pp. 16–22.

[34] C. Li, A. Raghunathan, and N. K. Jha, "Improving the trustworthiness of medical device software with formal verification methods," *IEEE Embedded Syst. Lett.*, vol. 5, no. 3, pp. 50–53, Sep. 2013.

[35] W. Burleson, S. S. Clark, B. Ransford, and K. Fu, "Design challenges for secure implantable medical devices," in *Proc. Design Autom. Conf.*, 2012, pp. 12–17.

[36] V. Pournaghshband, M. Sarrafzadeh, and P. Reiher, "Securing legacy mobile medical devices," in *Proc. Int. Conf. Wireless Mobile Commun. Healthcare*, 2012.

[37] C. Israel and S. Barold, "Pacemaker systems as implantable cardiac rhythm monitors,"

Amer. J. Cardiol., vol. 88, no. 4, pp. 442–445, Aug. 2001.

[38] K. Fotopoulou and B. Flynn, "Optimum antenna coil structure for inductive powering of passive RFID tags," in *Proc. IEEE Int. Conf. RFID*, Mar. 2007, pp. 71–77.

[39] G. P. Hancke and S. C. Centre, "Eavesdropping attacks on high-frequency RFID tokens," in *Proc. Workshop RFID Security*, Jul. 2008, pp. 100–113.

[40] *What is NFC Technology*. [Online]. Available: http://www.rfid-nfc.eu

[41] E. Haselsteiner and K. Breitfuss, "Security in near field communication," in Proc. Workshop RFID Security, Jul. 2006, pp. 3–13.

[42] H. Baldus, S. Corroy, A. Fazzi, K. Klabunde, and T. Schenk, "Human-centric connectivity enabled by body-coupled communications," *IEEE Commun. Mag.*, vol. 47, no. 6, pp. 172–178, Jun. 2009.

[43] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *Proc. ACM Conf. Comput. Commun. Security*, Nov. 2009, pp. 410–419.

[44] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[45] M. Zhang, M. M. Kermani, A. Raghunathan, and N. K. Jha, "Energy-efficient and secure sensor data transmission using encompression," in *Proc. Int. Conf. VLSI Design*, Jan. 2013, pp. 31–36.

[46] Federal Information Processing Standards (FIPS), *Announcing the Advanced Encryption Standard (AES)*, Nov. 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[47] National Institute of Standards and Technology (NIST), *SHA-3 Selection Announcement*. [Online]. Available: http://csrc.nist.gov/groups/ST/hash/sha-3/sha-3_selection_announcement.pdf

[48] T. Moore, "IEEE 802.11-01/610r02: 802.1.x and 802.11 key interactions," Microsoft Res., Tech. Rep., 2001.

[49] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, 2008, pp. 128–139.

[50] S. Jana *et al.*, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Proc. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 321–332.

[51] S. T. Ali, V. Sivaraman, and D. Ostry, "Zero reconciliation secret key generation for body-worn health monitoring devices," in *Proc. ACM Conf. Security Privacy Wireless Mobile Netw.*, 2012, pp. 39–50.

[52] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li, "IMDGuard: Securing implantable medical devices with the external wearable guardian," in *Proc. IEEE Int. Conf. Comput. Commun.*, Apr. 2011, pp. 1862–1870.

[53] C. Cornelius *et al.*, "Who wears me? Bioimpedance as a passive biometric," in *Proc. USENIX Workshop Health Security Privacy*, Aug. 2012.

[54] S. Schechter, "Security that is meant to be skin deep: Using ultraviolet micropigmentation to store emergency-access keys for implantable medical devices," Microsoft Res., Tech. Rep. MSR-TR-2010-33, Apr. 2010.

[55] T. Denning, K. Fu, and T. Kohno, "Absence makes the heart grow fonder: New directions for implantable medical device security," in *Proc. Conf. Hot Topics Security*, Jul. 2008, pp. 1–7.

[56] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: Non-invasive security for implantable medical devices," in *Proc. ACM Conf. Special Interest Group Data Commun.*, Aug. 2011.

[57] M. Zhang, A. Raghunathan, and N. K. Jha, "MedMon: Securing medical devices through wireless monitoring and anomaly detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 6, pp. 871–881, Dec. 2013.

[58] K. S. Kwak, S. Ullah, and N. Ullah, *An Overview of IEEE 802.15.6 Standard*, 2011. [Online]. Available: arxiv.org/pdf/1102.4106

[59] P. Li and R. Bashirullah, "A wireless power interface for rechargeable battery operated medical implants," *IEEE Trans. Circuits Syst.*, vol. 54, no. 10, pp. 912–916, Oct. 2007.

[60] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green, "Energy harvesting from human and machine motion for wireless electronic devices," *Proc. IEEE*, vol. 96, no. 9, pp. 1457–1486, Sep. 2008.

[61] *Infected Apps Found in Google's Android Market*, 2011. [Online]. Available: http://www.securitynewsdaily.com

[62] iPhone malware paradigm, 2012. [Online]. Available: http://www.crosstalkonline.org

[63] S. W. Smith and S. Weingart, "Building a high-performance, programmable secure coprocessor," *Comput. Netw.*, vol. 31, pp. 831–860, Apr. 1999.

[64] ARM, *Trustzone Technology Overview*. [Online]. Available: http://www.arm.com/products/CPUs/arch-trustzone.htm

[65] Trusted Computing Group. [Online]. Available: https://www.trustedcomputinggroup.org

[66] N. Aaraj, A. Raghunathan, and N. K. Jha, "Analysis and design of a hardware/software trusted platform module for embedded systems," *ACM Trans. Embedded Comput. Syst.*, vol. 8, pp. 8:1–8:31, Jan. 2009.

[67] J. M. Sorber, M. Shin, R. Peterson, and D. Kotz, "Plug-n-Trust: Practical trusted sensing for mHealth," in *Proc. Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 309–322.

[68] J. Sorber *et al.*, "An amulet for trustworthy wearable mHealth," in *Proc. Workshop Mobile Comput. Syst. Appl.*, 2012, pp. 7:1–7:6.

[69] C. Cornelius *et al.*, "Who wears me? Bioimpedance as a passive biometric," in *Proc. USENIX Workshop Health Security Privacy*, Aug. 2012.

[70] C. Cornelius and D. Kotz, "Recognizing whether sensors are on the same body," in *Proc. Int. Conf. Perv. Comput.*, 2011, pp. 332–349.

[71] N. Aaraj, A. Raghunathan, and N. K. Jha, "Virtualization-based framework for malware defense," in *Proc. Conf. Detection Intrusions Malware Vulnerability Assessment*, Jul. 2008, pp. 64–87.

[72] D. Arora, S. Ravi, A. Raghunathan, and N. K. Jha, "Secure embedded processing through hardware-assisted run-time monitoring," in *Proc. Design Autom. Test Eur. Conf.*, Mar. 2005, vol. 1, pp. 178–183.

[73] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Int. Cryptology Conf.*, Aug. 1999, pp. 388–397.

[74] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, other systems," in *Proc. Int. Cryptology Conf.*, Aug. 1996, vol. 1109, pp. 104–113.

[75] D. Agrawal, B. Archambeault, and J. R. Rao, "The EM side-channel(s)," in *Proc. Workshop Cryptogr. Hardware Embedded Syst.*, Aug. 2002, pp. 29–45.

[76] D. Kune *et al.*, "Ghost talk: Mitigating EMI signal injection attacks against analog sensors," in *Proc. IEEE Symp. Security Privacy*, May 2013.

[77] M. A. Hasan, "Power analysis attacks and algorithmic approaches to their countermeasures for Koblitz curve cryptosystems," *IEEE Trans. Comput.*, vol. 50, no. 10, pp. 1071–1083, Oct. 2001.

[78] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," in *Proc. Eur. Solid-State Circuits Conf.*, 2002, pp. 403–406.

[79] K. Tiri and I. Verbauwhede, "Charge recycling sense amplifier based logic: Securing low power security ICs against DPA," in *Proc. Eur. Solid-State Circuits Conf.*, 2004, pp. 179–182.

[80] R. Muresan and S. Gregori, "Protection circuit against differential power analysis attacks for smart cards," *IEEE Trans. Comput.*, vol. 57, no. 11, pp. 1540–1549, Nov. 2008.

[81] G. B. Ratanpal, R. D. Williams, and T. N. Blalock, "An on-chip signal suppression countermeasure to power analysis attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 3, pp. 179–189, Jul.-Sep. 2004.

[82] P.-C. Liu, H.-C. Chang, and C.-Y. Lee, "A low overhead DPA countermeasure circuit based on ring oscillators," *IEEE Trans. Circuits Syst.*, vol. 57, no. 7, pp. 546–550, Jul. 2010.

[83] S. Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos, and Y. Xie, "Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 64–69.

[84] M. Bucci, R. Luzzi, M. Guglielmo, and A. Trifiletti, "A countermeasure against differential power analysis based on random delay insertion," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 3547–3550.

[85] M. Zhang and N. K. Jha, "FinFET-based power management for improved DPA resistance with low overhead," *ACM J. Emerging Technol. Comput. Syst.*, vol. 7, no. 3, pp. 10:1–10:16, Aug. 2011.

ABOUT THE AUTHORS

Meng Zhang received the B.S. degree in electronics from Beijing University, Beijing, China, in 2008 and the M.A. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2010, where he is currently working toward the Ph.D. degree in electrical engineering.

His research interests include low-power system design, computer security, and body area networks.

Anand Raghunathan (Fellow, IEEE) received the B.Tech. degree in electrical and electronics engineering from the Indian Institute of Technology, Madras, India and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA.

He is a Professor in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. Previously, he was a Senior Research Staff Member at NEC Laboratories America, Princeton, NJ, USA, where he led research projects related to system-on-chip architectures, design methodologies, and heterogeneous parallel computing. He has coauthored a book *High-Level Power Analysis and Optimization* (New York, NY, USA: Springer-Verlag, 1998) and eight book chapters, holds 21 U.S. patents, and has presented several invited talks and conference tutorials. He has published over 200 journal and conference papers, and received eight best paper awards and four best paper nominations.

Prof. Raghunathan has been a member of the technical program and organizing committees of several leading conferences and workshops. He has chaired the ACM/IEEE International Symposium on Low Power Electronics and Design, the IEEE VLSI Test Symposium, the IEEE International Conference on VLSI Design, and the ACM/IEEE International Conference on Compilers, Architecture, and Synthesis for Embedded Systems. He has served as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, *ACM Transactions on Design Automation of Electronic Systems*, the IEEE TRANSACTIONS ON MOBILE COMPUTING, *ACM Transactions on Embedded Computing Systems*, IEEE DESIGN & TEST OF COMPUTERS, and the *Journal of Low Power Electronics*. He was a recipient of the IEEE Meritorious Service Award (2001) and Outstanding Service Award (2004). He is a Golden Core Member of the IEEE Computer Society. He received a Patent of the Year Award (an award recognizing the invention that has achieved the highest impact), and two Technology Commercialization Awards from NEC. He was chosen by MIT's *Technology Review* among the TR35 (top 35 innovators under 35 years, across various disciplines of science and technology) in 2006, for his work on "making mobile secure."

Niraj K. Jha (Fellow, IEEE) received the B.Tech. degree in electronics and electrical communication engineering from Indian Institute of Technology, Kharagpur, India, in 1981, the M.S. degree in electrical engineering from the State University of New York (SUNY) at Stony Brook, Stony Brook, NY, USA, in 1982, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 1985.

He is a Professor of Electrical Engineering at Princeton University, Princeton, NJ, USA. He has coauthored or coedited five books titled *Testing and Reliable Design of CMOS Circuits* (Norwell, MA, USA: Kluwer, 1990), *High-Level Power Analysis and Optimization* (Norwell, MA, USA: Kluwer, 1998), *Testing of Digital Systems* (Cambridge, U.K.: Cambridge Univ. Press, 2003), *Switching and Finite Automata Theory* (Cambridge, U.K.: Cambridge Univ. Press, 2009, 3rd ed.), and *Nanoelectronic Circuit Design* (New York, NY, USA: Springer-Verlag, 2010). He has also authored 12 book chapters. He has authored or coauthored more than 390 technical papers. He has coauthored 14 papers, which have won various awards. He has received 14 U.S. patents. His research interests include FinFETs, low-power hardware/software design, computer-aided design of integrated circuits and systems, digital system testing, quantum computing, and secure computing. He has given several keynote speeches in the area of nanoelectronic design and test.

Prof. Jha is a Fellow of the Association for Computing Machinery (ACM). He has served as the Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS and an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and the *Journal of Electronic Testing: Theory and Applications*. He is currently serving as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS, the *Journal of Low Power Electronics*, and the *Journal of Nanotechnology*. He has also served as the Program Chairman of the 1992 Workshop on Fault-Tolerant Parallel and Distributed Systems, the 2004 International Conference on Embedded and Ubiquitous Computing, and the 2010 International Conference on VLSI Design. He has served as the Director of the Center for Embedded System-on-a-chip Design funded by the New Jersey Commission on Science and Technology. He is the recipient of the AT&T Foundation Award and the NEC Preceptorship Award for research excellence, the NCR Award for teaching excellence, and the Princeton University Graduate Mentoring Award. His publications won the Best Paper Award at ICCD'93, FTCS'97, ICVLSID'98, DAC'99, PDCS'02, ICVLSID'03, CODES'06, ICCD'09, and CLOUD'10. A paper of his was selected for "The Best of ICCAD: A collection of the best IEEE International Conference on Computer-Aided Design papers of the past 20 years," two papers by IEEE MICRO MAGAZINE as one of the top picks from the 2005 and 2007 Computer Architecture conferences, and two others as being among the most influential papers of the last ten years at IEEE Design Automation and Test in Europe Conference. He has coauthored another six papers that have been nominated for best paper awards. He has served on the program committees of more than 140 conferences and workshops.