PROTECTING LEGACY MOBILE MEDICAL DEVICES USING A WEARABLE SECURITY DEVICE

Vahab Pournaghshband[1] and Peter Reiher[2]

[1]Computer Science Department, University of San Francisco, San Francisco, USA

[2]Computer Science Department, University of California, Los Angeles, Los Angeles, USA

ABSTRACT

The market is currently sated with mobile medical devices and new technology is continuously emerging. Thus, it is costly, and in some cases impractical, to replace these devices for new ones with greater security. In this paper, we present the implementation of a prototype for Personal Security Device—a self-contained, specialized wearable device that augments security to existing mobile medical devices. The main research challenge for, and hence the state of the art of, the proposed hardware design is that the device, to work with legacy devices, must require no changes to either the medical device or its monitoring software. This requirement is essential since we aim to protect already existing devices, as making modifications to the device or its proprietary software often impossible or impractical (e.g., closed source executables and implantable medical devices). Through performance evaluation of this prototype, we confirmed the feasibility of having a special-purpose hardware with limited computational and memory resources to perform necessary security operations.

KEYWORDS

Wireless medical device security, Man-in-the-middle attack

1. INTRODUCTION

Mobile medical devices are vast in application and popularity, and are becoming increasingly worn by technology users [15,17]. For instance, studies show over 500 million people have already been using mobile health applications [1]. There were approximately 245,000 insulin pump users in 2005, and the market for insulin pumps grew at a rate of 9% from 2009 to 2016 [2]. Hanna et al. reports that in the U.S. alone, there are 25 million people with wireless implantable medical devices (IMD), and about 300,000 of these IMDs are implanted every year [5]. While the need for secure mobile medical systems is widely recognized [4, 3, 6, 7], many manufacturers have not addressed the security risks of such devices, and thus have provided little security for either the devices themselves, or for the data they create and transmit. As such devices are adopted for a greater range of uses and users, a larger pool of information will be threatened. It is vital that security measures are developed and implemented to prevent more damaging leaks. As a whole, the field of security is vital to protect users who would otherwise find themselves vulnerable to a plethora of attacks. In a technologically dependent world, security must be placed at the backbone of innovation. Communications security is one critical aspect of protecting mobile medical devices. These devices typically communicate to an intermediate computer that forwards its signals to a healthcare facility. Since such devices are typically used with little or no configuration by a user or healthcare provider, there is ample opportunity for attackers to mislead the device into communicating with a hacker's machine instead of its intended intermediary. The communication between the device and its intermediary (real or malicious) typically is wireless, making it more susceptible to

Natarajan Meghanathan et al. (Eds) : ACITY, AIAA, DPPR, CNDC, WIMNET, WEST, ICSS - 2019
pp. 161-170, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91716

eavesdropping and injections. A range of existing wearable medical devices are susceptible to man-in-the-middle attacks. In particular, many legacy devices that are already in use by millions of users were introduced with weak, or no authentication mechanisms, as default factory settings. We specifically address legacy mobile medical devices. These are existing already-in-market devices that have no, or very weak security mechanisms in place, compromising the privacy of millions of users who are already using them. Such devices are also expensive or impractical to replace with new versions that have necessary and strong security features. For instance, replacing a pacemaker would require surgery. While integrating better security into every wearable device would clearly be the best approach, we can still improve the security of such devices and the overall systems they fit into without rebuilding or altering them.

In response, in our prior work [14], we introduced the concept of Personal Security Device (PSD) to improve security for mobile medical system. In [18], we implemented the proposed PSD concept as a smartphone app to protect mobile medical device users. For many average users a smartphone app is significantly more attractive than wearing a specialized wearable device that is primarily used for security purposes. However, there are trade-offs in choosing this option. Here, we address numerous fundamental limitations and challenges associated with implementing a personal security device on the Android platform:

- PSD as an Android app suffers from having to compete for resources with other applications running on the device. For our purposes, this could mean having to delay the transmission of medical data. Similarly, a smartphone app could have delayed data caused by the Bluetooth or WiFi stack having non-security packets being processed instead of the security controlling packet.
- A typical smartphone is limited to the radio technologies of GPS, Bluetooth, and WiFi. However, some wearable devices use other radio technologies. For instance, medical devices use other radio technologies, such as Medical Implant Communication Service (MICS). The security features available through a smartphone app are, thus, not extendable to medical devices that use these technologies.
- Since there is only a single WiFi adapter, and a single Bluetooth adapter on most commercial smartphones, we would be limited in the number of mobile medical devices and their possible corresponding access points (AP).
- As with any new code, there is a potential for security exploits. It is possible, for example, that by using the PSD smartphone application, the patient would be worse off due to someone hacking the Android device (e.g., through the web). In this scenario the patient could also be lulled into a false sense of security that may not have existed without a “security app.”

To eliminate these critical limitations of having the PSD as a smartphone app, we designed, built and evaluated a self-contained, specialized wearable device designed to only provide security to all mobile medical devices a user is carrying. In this paper, we present our prototype for such wearable device. Furthermore, we show that this embedded device, despite its limited computational and memory resources, is still able to perform necessary security operations such as encryption. Our prototype was built on Arduino [7]. Arduino is an open-source platform that consists of both a physical programmable circuit board (microcontroller) and a piece of software that runs on a computer. The software is used to write and upload computer code to the physical board.

The paper is organized as follows: Section 2 presents Related Work which is followed by Preliminaries in Section 3. Threat Model, Overall Design, Evaluation, and Future Work are presented in Sections 4, 5, 6, and 7, respectively. Section 8 concludes the paper.

2. RELATED WORK

More recently, the security of mobile medical devices has received ample attention for the acknowledged problem that it presents [3], [4], [6], [7]. There has been some work on demonstrating attacks against various mobile medical devices [24], [25]. Other research work has presented defensive approaches against these types of attacks [23], [24], [25], [27], [29]. Among those proposed defensive approaches, Amulet [28], Shield [23], and IMDGuard [29], all require a special-purpose third-party device to facilitate security. Also, Denning et al. [22] introduced a communication cloakers, a class of devices that would share secret keys with an implantable medical device (IMD). Communication cloakers act as a third-party mediator in the IMD's communications with external programmers. This device preserves IMD battery power by bearing the burden of the verification of incoming requests, and can be recharged easily. IMDGuard proposes changes in the design of future IMDs for a more secure system, not enabling compatibility with legacy devices,. Amulet, by design, does not work with existing devices since it requires changes to the existing mHealth system since it asks the medical sensor to verify that it is indeed the right Amulet before connecting. Shield is the only solution that is designed to work with existing and even already implanted IMDs by requiring no changes to the device. Shield receives and jams an IMD's messages at the same time to prevent other listeners from decoding those messages. This approach protects the IMD so that only the authorized intermediary is able to decode them. Moreover, Shield protects the patient by jamming unauthorized commands. However, the idea behind Shield may not be practical for many mobile medical devices that operate on widely-used radio technologies such as Bluetooth or 802.11, because of the potential legal ramifications of jamming those signals and the nature of the radio technologies themselves. The most relevant project is an external monitor called MedMon [30]. MedMon is an external monitor which snoops on all radio frequency wireless communications to and from implantable or wearable medical devices (IWMD). Through anomaly detection means, MedMon identifies potentially malicious exchanges. When a potentially malicious transaction is detected, it is capable of reacting passively or actively. While MedMon protects the body-area network against integrity attacks by acting like a firewall, and against battery-draining attacks, it does not protect patient confidentiality and privacy.

3. PRELIMINARIES

3.1. Arduino Memory

In this section, we briefly describe the three different types of memories in a typical Arduino device and their purpose.

3.1.1. Flash Memory

In Arduino, flash memory is primarily used to store program image and any initialized data including the executables for the device. While it is permitted to execute code from flash, modifying data by an executing code is not possible. To modify the data, it must first be copied into SRAM.

3.1.2. SRAM

Static Random Access Memory (SRAM), can be read and written from the executing program. SRAM memory is used for several purposes by a running program, namely, static data, heap and stack.

3.1.3. EEPROM

EEPROM is a form of non-volatile memory that can be read or written from the executing program. EEPROM is used to store long-term information developed during the device's use. EEPROM I/O cost is considerably larger than SRAM. It has a finite lifetime of approximately 100,000 write cycles, while read cycles do not compromise the device life expectancy.

4. THREAT MODEL

In this Section, we present diverse threat models regarding mobile medical systems. This varies from ensuring the availability of medical and health devices, the integrity of data communication, and privacy issues posed through wearable device sensors' information leaks. The mobile medical device threat model encompasses several aspects. These devices can be exploited as tracking devices, compromising the user's privacy. Some wearable medical devices contain personal information about the user that, if exploited, can be exposed to unauthorized parties. In addition to capturing sensitive information entered onto health and medical systems, attacks using such systems can eavesdrop on users and their environment. The availability of the devices can be attacked by exhausting their often limited resources. For instance, an attacker may drain the battery by forcing the device to perform power-expensive and/or wasteful operations. This can be done by keeping the battery usage/power consumption of the device on the elevated state either by turning on power-expensive features and keeping them on, or by sending bogus messages that require the device to process them and create a response in an expensive set of operations. Memory, another scarce resource, can extend the attack surface. For instance, attacking the availability of critical medical devices such as pacemakers can lead to fatal consequences.

As illustrated in Figure 1, the mobile medical system threat model varies between devices. In the case of wearable medical devices, the consequences of attacks can be extreme, potentially allowing attackers to cause the devices to operate in a life-threatening manner. For example, consider a heart rate monitor carried by a patient that communicates via Bluetooth to the patient's home computer, which in turn, forwards heart rate data to the patient's doctor in real time. If an attacker can alter the data to fake a heart attack, the doctor may institute unnecessary emergency measures. To consider an even worse scenario, if the attacker were to conceal the actual signs of an impending heart attack, the doctor would be unaware of the need for immediate action.

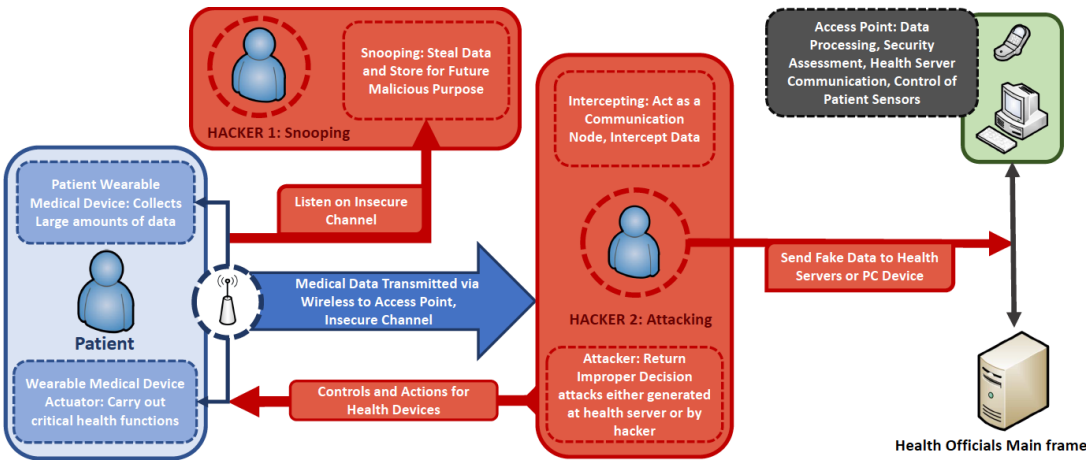

Figure 1. Threat Model and Potential Passive and Active Attacks on Mobile Medical Devices.

5. OVERALL DESIGN

5.1. Personal Security Device

We built a self-contained, specialized wearable device designed to only provide security to all mobile medical devices a user is carrying. High profile users, or those with particular security and privacy concerns are prime candidates for such a device. For instance, hacker might be interested in retrieving a presidential candidate's health information from a seemingly benign heart monitoring wearable device. While we built the PSD on an Arduino, we envision the PSD to be small, portable, inexpensive, and easy to use (Figure 2). It can be small enough to clip on a belt or fit in a pocket. With this approach, the medical device would still be simple by design whereas PSD is complex by design.

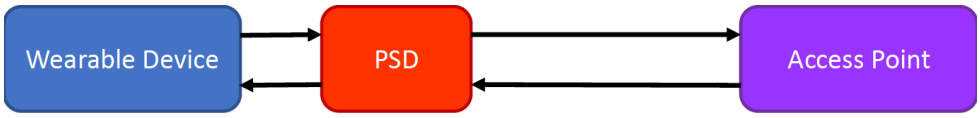

Figure 2. Illustration of the mobile medical device system when using the PSD.

The personal security device has two important roles:

1. The PSD is aware of the suite of wireless medical devices the owner uses, and it will have built-in knowledge of their security properties and vulnerabilities. The PSD will take steps to augment the security of the owner's devices, such as adding encryption to data streams, if the existing authentication is weak or absent. In addition, since the PSD is a device designed for security, it can provide various security features such as key or secure traffic management between wearable devices and systems they communicate with (such as an access point, smartphone, or other mobile medical devices).

2. The PSD will monitor the environment in which the owner operates, looking for known attacks and suspicious events. For instance, the PSD could watch for attempts by devices to reconfigure mobile medical devices suspiciously and raise alerts. Further, the PSD is itself a mobile device that relies on battery power, and must operate for extended periods without recharging; this puts limits on the processing, memory, and networking. Sophisticated analysis algorithms will not be feasible for the PSD. For these reasons, the device should perform anomaly detection whether on a more powerful and resourceful device, or on another device that it is connected to (e.g., cloud).

Furthermore, the PSD to work with existing mobile medical devices should have the following characteristics:

- No Changes to the Wearable Device: To secure existing wearable devices, the defense mechanism should not require any changes to the device.
- No Changes to the Monitoring Software: The defense mechanism should not require changes to the implementation of the proprietary monitoring software running on the access point (AP, a computer or a smartphone). This, coupled with the previous requirement for "no changes to the wearable device," would improve the security of the existing systems. Note that minor changes to the operating system running on the AP are still acceptable.
- Security vs. Availability: A robust defense mechanism should not decrease the functionality of the system. Also, it should not provide new avenues for an unauthorized person to drain a device's battery. Furthermore, the mechanism itself

should not introduce significant power or memory requirements that threaten the availability of the device itself.

5.2. PSD Prototype Specifications

Our prototype consists of an Arduino Mega 2560 with 16 analog input pins, 54 digital I/O pins, and a USB connection to power and upload sketches. In regard to memory, this Arduino has 256KB flash memory, 8KB SRAM, and 4KB EEPROM, which is an important design limitation we considered as we developed this device. The PSD includes a JY-MCU Bluetooth serial port module.

A shield is attached on top of the microcontroller to include additional memory, along with several other modules. The pertinent mounted modules for this prototype device are:

- an Adafruit Ultimate GPS Breakout – Version 3 with an internal patch antenna.
- a Liquid Crystal Display shield that is initialized in sketches with the pin ordering LiquidCrystal lcd (8,9,4,5,6,7).
- a Roving Networks RN-XV 171 WiFly module equipped with an 802.11 b/g radio.

We incorporated a standard AES library [20] in our implementation for our symmetric key encryption. We used 128, 192, and 256 bit key sizes.

6. EVALUATION

6.1. Performance Analysis

In this Section, we present memory requirements to perform necessary security operations by PSD in the Arduino specification presented in Section 5.2. Additionally, we measure the processing time it takes to complete those operations.

6.1.1. Memory Management

AES encryption requires 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Each round of processing includes one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. In our implementation, the AES encryption key is stored in EEPROM. SRAM will hold all temporary and intermediate results for the AES process. Furthermore, the encrypted blocks are stored in flash.

While performing the cryptographic operations, we measured the running time and the required memory to complete each operation: setting up the encryption key, encryption, and decryption. To measure the available memory in SRAM, we used the functions available in MemoryFree library [21] available for Arduino. Since only the encryption key is stored in EEPROM, 128-, 192-, and 256-bit keys require 16, 24, and 32 bytes in EEPROM respectively. Figure 3 and Table 1 summarize these measurements. Our findings confirm that it is feasible to perform adequate cryptographic operations, without exhausting the processing and memory resources of out PSD prototype.

Table 1. Processing time to perform AES operations on our proposed prototype when AES encryption keys are 128 bit, 192 bit, and 256 bit.

	Cryptographic Operation		
Key Size	**Key Setup**	**Encryption**	**Decryption**
128 bit	0.37 ms	0.58 ms (27.5 KB/s)	0.77 ms (20.5 KB/s)
192 bit	0.41 ms	0.71 ms (22.5 KB/s)	0.92 ms (17.5 KB/s)
256 bit	0.52 ms	0.82 ms (19.5KB/s)	1.09 ms (14.5 KB/s)

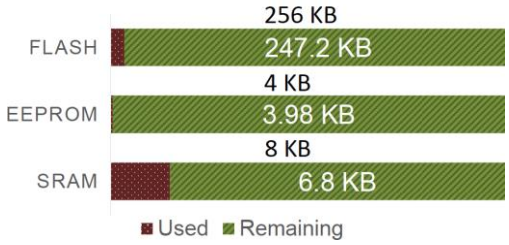

Figure 3. Memory used to perform cryptographic operations.

7. FUTURE WORK

7.1. Authentication and Key Generation

Data captured by wearable systems can enable new approaches to authentication. Smartphone sensor data has been used for authentication via differences in tap timing [8] and bioimpedance [9]. Examples of mobile biometric authentication include the use of gait [10] [11], walking patterns [12], and gaze [13]. Leveraging this data coupled with hardware-based approaches to implementing security protocols may be considered. One direction involves authentication using air handwriting analysis and lip movement. The PSD can be used to facilitate and manage hybrid authentication systems. In addition, the PSD can use this pool of information as seeds for key-generation for cryptographic protocols, when they appear to be random.

7.2. Fail-Open vs. Fail-Safe

PSD enables both a fail-open or fail-safe approach for wearable devices. While most wearable devices are more secure with a fail-safe approach, some must have the fail-open property. For instance, wearable medical devices have the unique property that they must fail-open when unbounded access is needed in emergency situations. In other words, security in life-critical medical devices should never come ahead of accessibility. For instance, if a patient with an implantable defibrillator collapses, the treating doctor would need to be able to communicate with the device to retrieve the patient's information and history, and issue necessary commands for treatment. Denying access to the doctor in such a situation is unacceptable.

7.3. Usability

Thorough and extensive usability studies need to be conducted, such as: the length of time needed for wearable device users of different ages to properly use this specialized device. This study must include people to be chosen with different technical background, as this application should be easy for someone with little technical background to operate. Perhaps more

importantly, this study should target the elderly, as they are likely candidates for this device. Further, usability studies must include those with limited mobility or arthritis, as these conditions would likely be prevalent in those needing assistive medical devices. Additionally, the tradeoff between security and usability and the usage of software versus special purpose hardware to secure wearable devices, will also be considered.

8. CONCLUSIONS

Mobile medical devices still pose a great risk to their users due to their use of an overly trusting procedure to communicate with their desired access point. This risk can be moderate, as is the case of an attacker stealing data from a pulse oximeter, to severe, as is the case of an adversary sending unauthorized malicious commands to a pacemaker. These attacks can be mitigated through a self-contained, specialized wearable device—Personal Security Device. We built a prototype to investigate the feasibility of such device. Through performance analysis, we confirmed that all necessary security operations can be performed in such prototype with limited computational and memory resources.

ACKNOWLEDGEMENTS

The authors are grateful to Majid Sarrafzadeh, Priyansha Gupta, and Karen Truong for their contributions. This work is supported by the National Science Foundation, CNS-1116371.

REFERENCES

[1] Mobile Health Apps: What Do You Use? http://www.cbc.ca/news/pointofview/2010/12/mobile-health-apps-what-do-you-use.html.

[2] Insulin pumps - global pipeline analysis, opportunity assessment and market forecasts to 2016, globaldata. *Global Data* (2010).

[3] Avancha, Sasikanth, Baxi, Amit, & Kotz, David. (2012) “Privacy in mobile technology for personal healthcare”. *ACM Computing Surveys*.

[4] Halperin, Daniel, Tadayoshi Kohno, Thomas Heydt-Benjamin, Kevin Fu, & William H. Maisel. (2008) “Security and privacy for implantable medical devices”, *Pervasive Computing.*

[5] Hanna, Kathi E. (2001) *Innovation and invention in medical devices: workshop summary*. National Academies Press.

[6] Kotz, David. (2011) “A threat taxonomy for mHealth privacy”, In *2011 Third International Conference on Communication Systems and Networks* (COMSNETS 2011), IEEE, pp. 1–6.

[7] Maisel, William H. (2005) “Safety issues involving medical devices”, JAMA: the journal of the American Medical Association, Vol. 294, No. 8, pp. 955–958.

[8] Giuffrida, Cristiano, Kamil Majdanik, Mauro Conti, Herbert Bos, (2014) “I Sensed It Was You: Authenticating Mobile Users with Sensor-enhanced Keystroke Dynamics.”, In *Proceedings of the 11th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2014).*

[9] Cornelius. Cory, Jacob Sorber, Ronald Peterson, Joe Skinner, Ryan Halter, & David Kotz. (2012) “Who wears me? bioimpedance as a passive biometric”, In *Proceedings of the 3rd USENIX conference on Health Security and Privacy (HealthSec'12).*

[10] Hong. Lu, Jonathan Huang, Tanwistha Saha, & Lama Nachman. (2014) "Unobtrusive gait verification for mobile phones", In *Proceedings of the 2014 ACM International Symposium on Wearable Computers (ISWC '14).*

[11] Muaaz, Muhammad & Rene Mayrhofer. (2014) "Orientation Independent Cell Phone Based Gait Authentication", In *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia (MoMM '14).*

[12] Casale, Pierluigi, Oriol Pujol, and Petia Radeva. (2012) "Personalization and user verification in wearable systems using biometric walking patterns", *Personal Ubiquitous Comput. 16*, No. 5, pp. 563-580.

[13] Kumar, Manu, Tal Garfinkel, Dan Boneh, & Terry Winograd. (2007) "Reducing shoulder-surfing by using gaze-based password entry", In *Proceedings of the 3rd symposium on Usable privacy and security (SOUPS '07).*

[14] Pournaghshband, Vahab, Majid Sarrafzadeh, & Peter. L. Reiher, (2012) "Securing legacy mobile medical devices, in 3rd International Conference on Wireless Mobile Communication and Healthcare", MobiHealth12.

[15] Abramovich. Giselle, "15 mind-blowing stats about wearable technology," http://www.cmo.com/articles/2014/6/16/Mind Blowing Stats Wearable Tech.html, accessed: 2014-12-19.

[16] S. Bogaty, (2014) "Wearable tech device awareness surpasses 50 percent among us consumers, according to npd," https://www.npd.com/wps/portal/npd/ us/news/press-releases/wearable-tech-device-awareness-surpasses-50 percent-among-us-consumers-according-to-npd/, accessed: 2014-12 26.

[17] Comstock, Jonah, (2014) "Pwc: 1 in 5 americans owns a wearable, 1 in 10 wears them daily," http://mobihealthnews.com/37543/pwc-1-in-5-americansowns-a-wearable-1-in-10-wears-them-daily/, accessed: 2014-12-19.

[18] Pournaghshband, V., Sarrafzadeh, M., and Reiher, P., (2014) "Adrasteia: A Smartphone App for Securing Legacy Mobile Medical Devices," In Proceedings of IEEE Workshop on Usable Mobile Security (WUMS).

[19] Arduino Project. https://www.arduino.cc/

[20] AES Encryption library. http://utter.chaos.org.uk:/~markt/AES-library.zip

[21] Arduino MemoryFree library. https://github.com/maniacbug/MemoryFree

[22] Denning, Tamara, Kevin Fu, & Tadayoshi Kohno, (2008) "Absence makes the heart grow fonder: New directions for implantable medical device security", In *Preceedings of the 3rd Conference on Hot topics in security*, USENIX Association, p. 5.

[23] Gollakora, Shyamnath, Haitham Hassanieh, Benjamin Ransford, Dina Katabi, & Kevin Fu, (2011) "They can hear your heartbeats: non-invasive security for implantable medical devices", In *Proc. of the ACM SIGCOMM Computer Communication Review.* pp. 2-13.

[24] Halperin, Daniel, Thomas S. Heydt-Benjamin, Benjamin Ransford, Shane S. Clark, Benessa Defend, Will Morgan, Kevin Fu, Tadayoshi Kohno, & William H. Maisel, (2008) "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," *in 2008 IEEE Symposium on Security and Privacy* IEEE, pp. 129–142.

[25] Li, Chunxiao, Anand Raghunathan, & Niraj K. Jha, (2011) "Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system", *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*, pp. 150–156.

[26] Ott, Len. (2010) "The evolution of Bluetooth in wireless medical devices", *Socket Mobile, Inc. White Papers*.

[27] Rasmussen, Kasper Bonne, Claude Castelluccia, Thomas S. Heydt-Benjamin, & Srdjan Capkun, (2009) "Proximity-based access control for implantable medical devices", *In Proc. of 16th ACM Conference on Computer and Communications security*.

[28] Sorber, Jacob, Minho Shin, Ronald Peterson, Cory Cornelius, Shrirang Mare, Arathi Prasad, Zachary Marois, Emma Smithayer, & David Kotz. (2012) "An amulet for trustworthy wearable mHealth", In *HotMobile* (New York, NY, USA, 2012), ACM, pp. 7:1–7:6.

[29] Xu, Fengyuan., Zhengrui Qin, Chiu C Tan, Baosheng Wang, & Qun Li, (2011) "IMDguard: Securing implantable medical devices with the external wearable guardian", In *INFOCOM*.

[30] Meng. Zhang, Anand Raghunathan, & Niraj K Jha, (2013) "Medmon: Securing medical devices through wireless monitoring and anomaly detection," *Biomedical Circuits and Systems, IEEE Transactions on*, Vol. 7, No. 6, pp. 871–881.

Authors

Vahab Pournaghshband is an Assistant Professor in the Computer Science department at University of San Francisco. He received his Ph.D. from the Computer Science department at University of California, Los Angeles (UCLA) in 2014. He received his M.Sc. in Computer Science from University of California, Berkeley. Also from UC Berkeley, he received his B.Sc. in Electrical Engineering and Computer Science. Dr. Pournaghshband has done research in the fields of computer networks, computer security, and computer science education.

Peter Reiher received his B.S. in Electrical Engineering and Computer Science from the University of Notre Dame in 1979. He received his M.S. and Ph.D. in Computer Science from UCLA in 1984 and 1987, respectively. He has done research in the fields of distributed operating systems, network and distributed systems security, file systems, ubiquitous computing, mobile computing, and optimistic parallel discrete event simulation. Dr. Reiher is an Adjunct Professor in the Computer Science Department at UCLA.