COMPUTERS & SECURITY XXX (XXXX) XXX

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Bulwark: Securing implantable medical devices communication channels

Lake Bu[a,1,*], *Mark G. Karpovsky*[b], *Michel A. Kinsy*[a]

[a] *Adaptive and Secure Computing Systems Laboratory, Department of Electrical and Computer Engineering, Boston University, Boston, USA*

[b] *Reliable Computing Laboratory, Department of Electrical and Computer Engineering, Boston University, Boston, USA*

ARTICLE INFO

Article history:
Received 28 May 2018
Revised 15 October 2018
Accepted 16 October 2018
Available online xxx

Keywords:
Implantable medical devices
Security
Man-In-The-Middle
Encryption
Authentication
Robust codes
AMD codes

ABSTRACT

Implantable medical devices (IMDs) have been used to manage a broad range of diseases and ailments. They are convenient for patients due to their small sizes, unobtrusiveness and portability using wireless monitors or controllers. However, the wireless communication between these devices and their controllers often lacks security features or mechanisms. This lack of security makes the use of these devices a fertile ground for passive and active attacks. Unlike other cyber attacks which target victims' information or property, attacks on medical devices can threaten a victim's life. Currently, there are very few efficient solutions to these attacks which balance security, reliability, and power consumption. Therefore, in this work, we propose a robust approach for guarding against existing and potential communication-based attacks on IMDs while keeping the added hardware and power consumption low. In addition, we introduce a secure and efficient protocol for authorizing third-party medical teams to access the IMDs in the case of an emergency.

1. Introduction

Implantable medical devices (IMDs) such as insulin pumps, pacemakers, and self-powered biosensors are widely used to save lives or improve quality of life. These devices are generally embedded inside patients' bodies and communicate through wireless transmissions with their controllers or monitors, depending on whether they are open-loop or closed-loop systems.

In the past decade, various attacks on the IMDs have been reported. Many of these attacks have been successful in either acquiring the health data or manipulating the functionality of the IMDs. Most of these attacks used a Man-In-The-Middle (MITM) attack approach and exploited the wireless communication infrastructure. Furthermore, since the medical data transmitted through the wireless channel are highly repetitive or in a regular pattern (such as heart beats or glucose in the blood), it is fairly straightforward to predict the information even if it is encrypted, which makes IMDs far more vulnerable to attacks.

Eavesdropping is one of the most commonly seen passive attacks on wireless channels. The attackers simply listen to the unencrypted transmissions acquire the health data of the targeted patients or victims. Since there is no malicious tampering of the transmission, it is hard to detect. There are software and hardware techniques to eavesdrop the IMDs' channel. In fact, several research efforts have investigated this particular class of passive attacks, e.g., Halperin et al. (2008); Li et al. (2011a), and Paul et al. (2011), etc.

* Corresponding author.
E-mail address: bulake@bu.edu (L. Bu).

[1] Part of the work was done while Lake Bu was at the Reliable Computing Laboratory at Boston University.

https://doi.org/10.1016/j.cose.2018.10.011
0167-4048/

Whereas eavesdropping only aims to steal the victim's medical information, active attacks such as hijacking or replay are more alarming because they can directly interfere with the victim's health and life. The attackers can use radio transmitters to generate commands to the devices implanted inside patients' bodies. They can either send their own forged commands, or replay a legal command eavesdropped and stored previously. These types of attacks have been studied by Halperin et al. (2008) on pacemakers and Roberts (2011) on insulin pumps. These attacks have been shown in simulation to result in fatal attacks.

Although, some IMD manufacturers have implemented encryption functions, e.g., Advanced Encryption Standard (AES), on their devices, they are often not activated due to power consumption concerns or authentication complexity when communicating with third party devices (InfoSec, 2014). Oftentimes, even if the encryption module is activated and the transmission is encrypted, the authentication part of the communication is neglected. These functionality-oriented design choices that trade security for lower power usage have made these medical devices highly vulnerable to attackers.

1.1. *Motivation and contribution*

As stated above, a secure and practical IMD transmission channel should satisfy the following properties:

- The confidentiality and integrity of the transmitted and received packets should be guaranteed;
- When all the security features are turned on, the power consumption overhead should be negligible (i.e., below 10%);
- A robust third party (e.g., emergency team or hospital) authentication scheme that strikes an effective balance between security and practicality.

Therefore, in this paper we extend our previous work (Bu and Karpovsky, 2017) on the subject and develop a new approach for the secure and reliable wireless transmission of data in IMD applications using authenticated encryption against both passive and active MITM attacks. The major contributions are:

- The transmitted messages are obfuscated against passive attacks such as eavesdropping, and timestamped and authenticated against active attacks such as replay, tampering or message forgery;
- Such protection only adds a small hardware and power overhead, specifically *less than 10%* of the design;
- We introduce a threshold-based third-party authorization protocol to be used in the event of a medical emergency where the medical device needs to be accessed, ensuring both availability and security.

The rest of the paper is organized as the following. Section 2 briefly explains the several IMD transmission models. Section 3 illustrates the existing and potential attacks against current IMDs. Section 4 defines the criteria of the protection against such attacks. Section 5 introduces the proposed protection scheme and its work flow, as well as the theoretical estimation of its security level. Section 6 presents an effective third party authentication protocol to be used in the case of a medical emergency. Section 7 evaluates the proposed design by experiments and overhead comparison with other schemes.

2. IMD communication models

There are various types of wireless IMDs currently in use. Generally, they are characterized by their functionality, deployment environment, communication protocol and power supply. Different attacks target different IMDs based on these characteristics.

2.1. *Closed-loop IMDs*

Closed-loop IMDs are self-monitored and self-managed. As shown in Fig. 1, they receive wireless transmissions from the sensor inside the patient's body and the actuator determines what therapy to deliver accordingly. The most commonly seen closed-loop IMDs are pacemakers and implantable cardiac defibrillators (ICDs) (Burleson et al., 2014).

Research has shown that many IMDs in this class have some form of encryption algorithm (e.g., AES) built into them, but due to power usage restrictions, the encryption modules are generally not turned on. Some of these devices are: Medtronic Maximo implantable cardiac defibrillator (ICD) series and BIOTRONIK Itrevia 7 DR-T/VR-T ICD series (Rostami et al., 2013). As a consequence, the data transmission on these devices is unencrypted, exposing the patient's raw sensor data. Eavesdropping and learning of the patient's medical information under these settings become fairly easy to carry out. Based on the acquired information, the attackers can replay some of the messages to the monitor, forcing the device to react in a certain way.

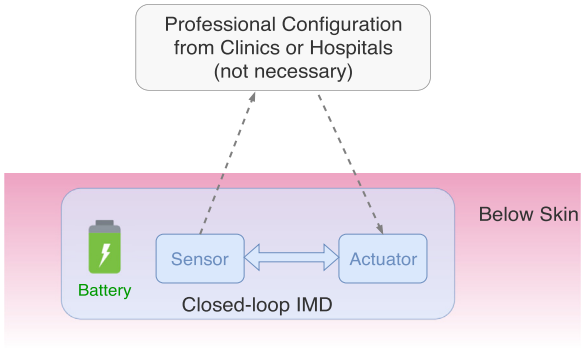

Fig. 1 – Closed-loop IMDs manage themselves based on the communication between the sensor and the actuator. Although they have no access for the patients to control them, they do allow configurations from professionals. The communication is generally not encrypted due to the power consumption consideration. The battery is usually not chargeable and the replacement requires surgery.

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

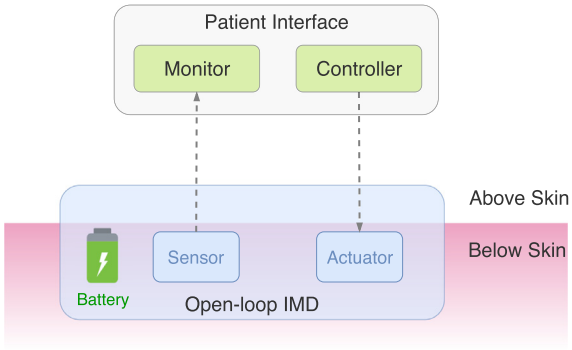

Fig. 2 – Open-loop IMDs usually come with a monitor and a controller. The patients monitor their health status based on the data from the sensor. They can then issue commands (such as an exact dosage of medicine or starting the insulin pump) based on the information provided. The communication is typically not encrypted.

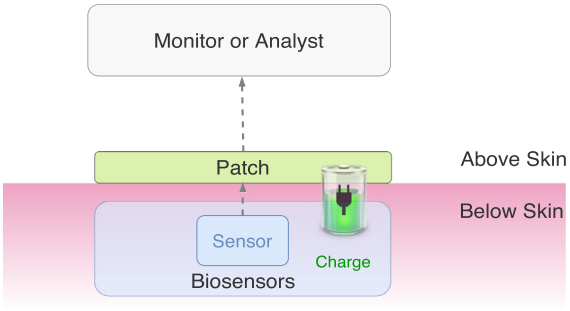

Fig. 3 – Biosensors send measurements to the patch and are powered by it. The patch then sends the measurement to a monitoring or analysis module.

Researchers (Halperin et al., 2008) highlighted some key security vulnerabilities that are present in current pacemakers and ICDs. They successfully listened to and interpreted the wirelessly transmitted information from the patient's device. Furthermore, they were even able to reuse stored messages to disable the device, which could cause fatal accidents.

Power usage constitutes a major design challenge for these IMDs, especially when it comes to their security and privacy features. Usually pacemakers or ICDs are designed to last for 5 to 7 years. Once the battery runs out of power, a surgery may be required to replace it. Hijacking the transmission channel by jamming or injecting data packets can also cause the device to operate in a high-power mode, resulting in a faster depletion of the battery.

2.2. *Open-loop IMDs*

Open-loop IMDs such as insulin pump systems can be more assailable. As shown in Fig. 2, they receive wireless transmissions from the devices' sensors inside patients, who are able to respond with remote controls. According to the reading of the patient's glucose levels, they may decide to inject themselves.

These IMDs' communications are generally not encrypted or authenticated (such as the Medtronic MM515/715 and MM523/723 series (Li et al., 2011b)), making them highly vulnerable to eavesdropping (*e.g.*, using simple software-defined radio tools) or forgery of malicious commands. Furthermore, most of the communicated messages are control signals, which allows even simple attacks to have harmful impacts on the victim.

For instance, the authors in Li et al. (2011a) examined insulin pump systems. They were not only able to acquire the encrypted information from the device, but also managed to forge false glucose readings at the monitor. In addition, they were successful at sending their own commands to the pump due to its lack of authentication. Moreover, other researchers such as Radcliffe (2011) and Takahashi (2011) showed that they were able to gain full control of some insulin pump systems, because the devices accept unauthorized radio signals or commands.

2.3. *Biosensors*

Biosensors are different from the two types explained above in several ways. As shown in Fig. 3, first, they are usually powered by a transmitter 152 or self-powered inside human bodies. Second, they are purely transmitters and receive no commands. Biosensors are widely used to detect glucose, lactate, or cholesterol etc. The transmitter, i.e., patch, serves as the middle station which powers the sensor while sending the data to a higher level monitor or analysis module. The monitor sends no data or commands to the patch or the biosensor.

The major threat to biosensors is eavesdropping. However, eavesdropping cannot be easily carried out because of the short communication distance between the patch and the monitor/analyst. Other precise and practical threat models are yet to be developed and demonstrated (Burleson et al., 2014).

3. Existing and potential attacks to IMDs

As mentioned in the previous section, for IMDs with wireless communication, eavesdropping and channel hijack are the two most frequently reported attacks. Although many IMDs are equipped with an encryption function, it often not enabled (cf. Section 2). Even if the encryption function is activated, it can only prevent the attackers from eavesdropping and understanding the patients' health information. Those devices may still be vulnerable to certain classes of active Man-In-The-Middle (MITM) attacks such as hijack and replay. In this work, we define weak and strong attack models along these lines.

Definition 3.1. The weak attack model is defined as follows:

1. The attacker is able to eavesdrop the victim's wireless IMD transmission between the sensor and the monitor/controller (Rostami et al., 2013);
2. The attacker is capable of using a programmed radio to interfere with the transmitted packets of the victim's actuator (Rushanan et al., 2014);

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

3. The attacker has no knowledge of the format of the data packets or the information (health data, commands etc.) they carry.

Definition 3.2. The strong attack model is defined as follows:

1. Same as in the weak attack model;
2. Same as in the weak attack model;
3. The attacker has the ability to acquire the format of the data packets, and make a reasonable estimation or prediction (Rushanan et al., 2014; Yury, 2014) of the information they carry.

For the strong attack model, although it is feasible to learn the IMD data packet format, it does require some effort. Usually, IMD manufacturers do not publish the message format or the command codes. Without the official instructions, the attacker needs to first learn the message formatting, and some IMDs have multiple distinct packet formats. However, with enough literature search, e.g., articles, patent filings, the attacker can gather sufficient information on the formats.

To learn about the transmission or transmitted data packets, the attacker needs some wireless peripheral equipment (e.g., Texas Instruments (TI) CC1101 RF Transceiver) to capture, buffer and analyze these packets near real-time. In some cases, the data transmission frequency can be challenging to learn or calculate (Radcliffe, 2011). Since in most IMDs, the transmission frequency is fairly low, *e.g.*, one read in five minutes for some insulin pumps.

Given the definitions above, the following sub-sections will describe the existing and potential attacks on unprotected and encryption protected IMDs.

3.1. *IMDs with disabled encryption*

As mentioned in Section 2, most IMDs have their encryption module (e.g., AES) disabled, leaving the transmission channel entirely unprotected. Once the attackers eavesdrop and analyze the transmitted messages, they are capable of applying various attacks such as replay or spoofing commands. Attacks can result in the leakage of patients' health information, increase of battery power consumption, overdose of the medicine, and malfunction or even termination of the implanted devices etc.

3.2. *IMDs with enabled encryption*

Even if the medical device has an active encryption mode and encrypts each transmitted message, it is still vulnerable to other classes of attack (InfoSec, 2014).

3.2.1. *Eavesdropping*

Eavesdropping is a type of passive attack in which the attacker silently listens to the unencrypted wireless transmission. The attacker does not necessarily apply any malicious modifications to the transmitted messages. Usually, the goal of eavesdropping is to acquire the victim's important health information or the device's transmission packets.

The Advanced Encryption Standard (AES) (Daemen and Rijmen, 2013) is a well-known solution that prevents attackers from understanding the message transmitted even if they record it. However, for an IMD usually the number of commands is very limited. Therefore the attackers may be able to predict or make a proper guess of the correlation between the cipher and plaintext.

3.2.2. *Hijack and replay*

As highlighted above, some IMDs have no mechanism in place to authenticate incoming radio signals. Thus, the attackers can establish and execute anonymous transmissions to either the implanted device or the monitor/controller. This vulnerability gives attackers an opportunity to take over the transmissions between legal sensors and controllers. An attacker can first eavesdrop and record a set of legal, encrypted, health data, message transmissions. Then, they can replay some of the stored legal encrypted messages back to the IMD.

Therefore, even if every transmitted, legal, message is encrypted and authenticated, a replay attach will still be successful. Moreover, if the attacker can conduct these replay attacks, they can choose or recompose a subset of the original legal (i.e., encrypted and authenticated) messages to harmfully affect the IMD. For example, very high glucose measurements can be frequently sent to the patient's monitor, inducing overdose of insulin (Rushanan et al., 2014). Commands of persistent large electric shocks can be sent to an ICD, causing a deadly health event, *e.g.*, a cardiac arrest.

The deployment mode of these devices also makes the implementation of encryption algorithms deemed cryptographically secure in their general form less secure. For example, if an IMD has AES with Cipher Block Chaining (CBC) mode turned on, the current plaintext can be randomized using the previous cipher-text before its encryption. Therefore, the decrypted messages would be beyond most attackers' capability. However, since the health data are usually generated using microprocessors and sensors of 8 bits, 12 bits, or 16 bits (Chede and Kula, 2008; McDonald et al., 2011), there is a high probability that a replay attack message will be interpreted and executed as a legal message.

Example 3.1. In this example we show that with replay of an intercepted cipher, an attacker is able to transmit erroneous data potentially harmful to the victim, even if they have no knowledge of the plaintext related to the replayed cipher.

For an insulin pump IMD, the acceptable glucose reading is in a small range (approximately from 100 mg/dL to 200 mg/dL), which forms the limited range of the plaintext transmitted between the sensor and pump. A replayed cipher message can be decrypted to a legal numeric value within this range at a relatively high chance.

Suppose in a wireless transmission of a 128-bit AES-CBC protected insulin pump IMD system, the 128-bit Initializer Vector (IV) used to resist chosen-plaintext attacks is:

$$IV = \{0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f\}$$

and a 128-bit secret key is:

$$key = \{0x60, 0x3d, 0xeb, 0x10, 0x15, 0xca, 0x71, 0xbe, 0x2b, 0x73, 0xae, 0xf0, 0x85, 0x7d, 0x77, 0x81\}$$

This insulin pump generates 16-bit measurement data of glucose in the blood. From a previous eavesdropping, the at-

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

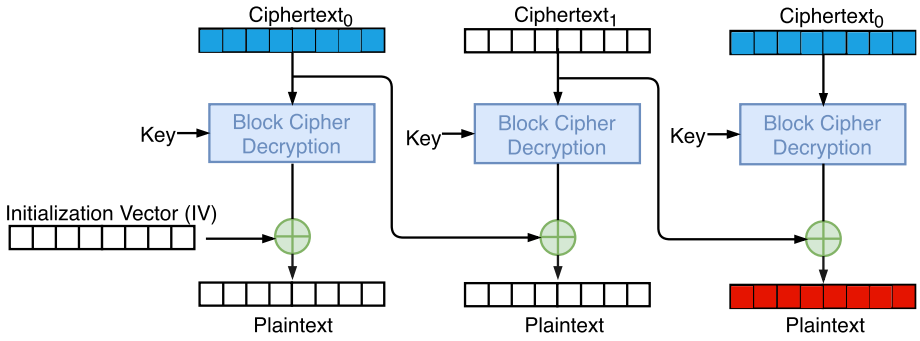

Fig. 4 – In the CBC mode, the decryption procedure is determined by both the current and previous ciphers. When the cipher at t_0 is replayed at time t_2 (blue), the decrypted plaintext can result in a fake but legal glucose reading. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

tacker has acquired the legal cipher of a sensor's measurement of glucose at moment t_0 as

$$cipher(t_0) = \{0x17, 0x71, 0x98, 0x42, 0xac, 0x9c, 0x9e, 0xe8, 0x87, 0xc6, 0xed, 0x71, 0xd1, 0x1a, 0x78, 0x24\}$$

After a meal at the moment t_1 the patient's IMD microprocessor transmits the cipher text for "200 mg/dL" high-glucose level glucose to his monitor:

$$cipher(t_1) = \{0x0e, 0x11, 0x43, 0x4e, 0x23, 0xb1, 0x32, 0xf2, 0x4c, 0x12, 0x0a, 0x6d, 0x2c, 0x03, 0x87, 0x1e\}$$

Then the attacker uses his own programmed radio to send the pre-stored $cipher_0$ soon after, although they have no knowledge of the plaintext that this cipher relates to. According to the CBC mechanism, with the secret key and the previous $cipher(t_1)$ the decryption module gets the following plaintext at moment t_2:

$$plaintext(t_2) = \{0x00, 0x8c, 0xe2, 0x41, 0xf2, 0x5f, 0x42, 0x07, 0x28, 0x59, 0x2a, 0x44, 0x52, 0xe2, 0x43, 0x5c\}$$

where the measurement bits are {0x00, 0x8c} which happens to be "140" at the *normal range*, resulting in a skip of medication.

The decryption procedure with the injected replay cipher is shown in Fig. 4.

Similar techniques also work for closed-loop devices such as ICDs and pacemakers.

3.2.3. *Bit-flipping attacks*

Another type of attack that takes advantage of the CBC mode is the bit(byte)-flipping approach. By maliciously flipping some of the bits in the previous cipher, the next decrypted plain text will be altered in exactly the same bits (Swepsie, 2014) as shown in Fig. 5.

Even if there is no leakage of the secret key, according to Definitions 3.1 and 3.2, as long as the attacker can listen to the channel and make a proper guess of the incoming message, the intrusion is most likely to succeed. Such attacks are usually applied to flip a bit in the IMD command message, which is more potent and can be fatal than distorting the sensor measurement data.

Example 3.2. In this example we illustrate how using a bit-flipping technique with the proper timing, a legal command can be converted to a harmful instruction in the IMD.

Suppose in a 128-bit AES-CBC protected insulin pump IMD system, the 128-bit IV and secret key are the same as Example 3.1. The system encodes and decodes 16-bit commands including issuing an injection {0x00, 0x80}, turning on the device {0x80, 0x00}, and turning off the device {0x08, 0x00} (Radcliffe, 2011).

After a patient finishes their meal and the glucose reading is high, the patient will be ready to issue the insulin injection command, {0x00, 0x80}. If the attacker can predict this event, they can simply inject a forged command cipher at moment t_1 as:

$$cipher(t_1) = \{0x08, 0x80, 0x35, 0xf6, 0x88, 0x28, 0x6e, 0xc1, 0x3a, 0xd0, 0x87, 0x60, 0x10, 0x90, 0xd5, 0xe0\}$$

It is worth noting that the attacker has some timing flexibility, i.e., t_1 just needs to be before or during the meal.

This forged cipher itself does not translate into any legal command. Thus, at t_1 no operation is carried out by the IMD. However, as the patient sends a following command to inject insulin, after decryption in the CBC mode at moment t_2 the command becomes.

$$plaintext(t_2) = \{0x00 \rightarrow 0x08, 0x80 \rightarrow 0x00, 0x60, 0x3d, 0xeb, 0x10, 0x15, 0xca, 0x71, 0xbe, 0x2b, 0x73, 0xae, 0xf0, 0x85, 0x7d\}$$

Which leads to the command to turn off the device instead of injecting insulin.

For closed-loop devices like ICDs and pacemakers, this type of attack is not straightforward, but it is still achievable. Since most IMDs only accept configurations from a clinic or a hospital, it requires that the attacker has access to an authorized professional configuration device or location. However, these professional configuration devices can be found on many third party medical device trading websites (MedWOW, 2018). With one of those configuration device, this bit-flipping attack becomes viable, and can lead to turning off a pacemaker and putting a patient in grave danger.

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

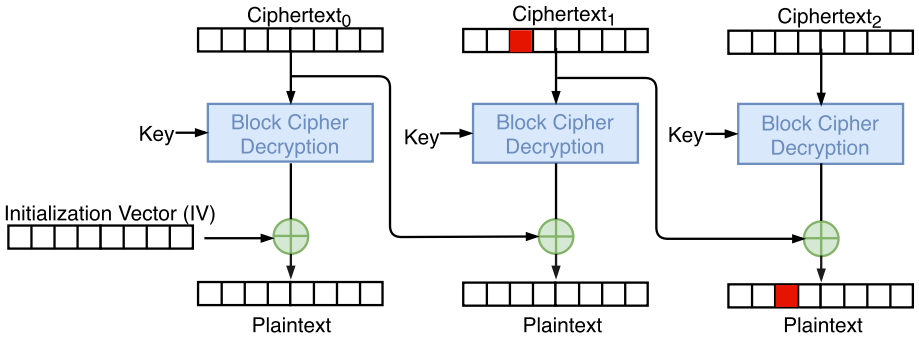

Fig. 5 – If a previous cipher is flipped by XOR operations in some bits, the next plaintext will be flipped accordingly by XOR operations in the same bits.

4. System design criteria

The goals of the proposed security system are to protect the IMD wireless channels from the MITM attacks mentioned above while maintaining a low power consumption overhead. In addition, this enhanced design methodology should only minimally impact the current approaches targeting secure IMD designs, so that it can be readily integrated in the existing IMD manufacturing processes.

4.1. *Eavesdropping resistant properties*

The general form of the AES algorithm is resilient against eavesdropping. Without the security key it is almost impossible to decrypt the messages within a reasonable time period.

However in the case of IMDs, it may not be sufficient. As stated in Definitions 3.1 and Definition 3.2, it is possible for an attacker to eavesdrop, store a number of legal ciphers, and learn from the encrypted data. The definition underlines the fact that an attacker may be able to properly guess the encrypted health data or commands (plaintext), since some of them appear in highly regular patterns. For example, the glucose level is usually between 70 to 200 mg/dL, and cardiac rhythms are known to be the biological signatures of each person. The command codes are also very limited. Since the plaintexts are predictable, they should be properly randomized. However, even if they are in the AES-CBC mode (by XORing the current health data or command bits with the previous cipher during encoding), the attackers still can analyze them since the ciphers have already been eavesdropped. Therefore, a more sophisticated randomization techniques is required.

4.2. *Hijack, replay, and bit-flipping attacks*

First, the transmission should be authenticated, so that unauthorized or replayed radio signals should not be accepted as legal sensor readings or commands. AES itself does not provide this feature and extra modules for authentication are required.

There are various authentication methods. Keyed-hash Message Authentication Code (HMAC) provides strong security but requires a large amount of extra bits for the digest, which contradicts the first design criteria by bringing considerable modification (each data packet uses more than one block) to the current IMD's security scheme. Here we propose to use the Robust codes (Bu and Karpovsky, 2016) or the Algebraic Manipulation Detection (AMD) (Wang and Karpovsky, 2011) codes which are both lightweight message authentication codes. Unlike the HMAC which has a fixed length of 160 bits, the two aforementioned codes are very flexible due to their support of variable data packet sizes.

The Robust codes are used against weak attacks as a keyless MAC code. With the definition of weak attacks, the attackers are not able to predict or make a close guess on the contents of transmitted packets. Therefore, their intrusions will be more of random error injections or bit flipping. In this case the Robust code provides a non-linear digest of the transmitted messages, which efficiently detects any malicious modification.

The AMD codes, however, have to deal with more severe cases where data packets can be intercepted and analyzed. The attackers' intrusions will be more targeted and carefully designed. With a close enough guess of the transmitted messages, the attackers will have a high probability of successfully injecting an error, as mentioned in the previous section. Thus the AMD codes bring in a random vector so that the plaintext is randomized even if the health data or commands are non-uniformly distributed, which efficiently resists the strong attacks.

Second, each valid cipher should only be authenticated once. Therefore, even if the attacker stores all the authenticated and encrypted transmissions, they will not be able to reuse any of them in the future. Thus it is necessary to use a self-incrementing timestamp or nonce in each transmission as part of the authentication process. The system always keeps track of the latest timestamp. If an incoming message has a timestamp smaller or equal to the highest one known by the system, it is illegitimate.

Medical devices tend to use low-frequency sensors with sampling rates from 1Hz to 1kHz and an IMD can remain functioning for 1 to 10 years. The security module should guarantee that within these years under the health data sampling rate, not a single replay or bit-flipping attack can succeed. Therefore based on these parameters, the attack misdetection probability should be at least 2^{-32} for IMDs working under low frequency of up to 10Hz and at least 2^{-40} for higher frequency of up to 1kHz.

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

4.3. Random error issues

Random errors are not attacks. They are usually caused by unstable transmissions or minor changes in voltage. Upon the presence of random errors, the readings of health data might be imprecise or the commands might be distorted. The reliability against random errors can be enhanced by applying error correction codes (ECC) to the plaintexts (Burleson et al., 2014). In this design we will use double error correction codes which is more than enough for the channel.

4.4. Data block size considerations

Many of current IMDs are equipped with AES using 128 or 192-bit encryption modes. The 128-bit mode is the most common used and the 256-bit mode is excessive in terms of both security and power consumption and thus much less deployed. Each piece of data or command packet is encrypted into a 128 or 192-bit data block. We aim to not increase the number of blocks or the size of blocks needed for each packet.

4.5. Power consumption

Since wireless IMDs are mostly battery powered (except for the self-powered biosensors), the design should also aim for low power consumption overhead compared to other possible methods.

5. The proposed secure wireless channel for IMDs

The proposed protection scheme takes MAC-then-Encrypt (MtE) as the order of protection. In this way the IMD's information part (health data from sensors or commands from controllers), the timestamp, and the authentication signature can be wrapped all under 128 bits or 192 bits depending on the demand. As a result, it adds no extra transmission overhead to the current IMDs equipped with 128 or 192-bit AES in CBC mode. Although MtE is not considered the most secure authenticated encryption mode, it has been proved to be secure with the AES-CBC mode (Krawczyk, 2001), which happens to be the case for most IMD devices.

5.1. Notations and concepts

To help describe and evaluate this protection mechanism, we introduce the following notations and concepts.

5.1.1. Finite field operators

We denote the Galois finite field by GF, and the numbers of bits in each data packet by b. Then $\cdot$ is the multiplication in the $GF(2^b)$ finite field, $\oplus$ the addition in $GF(2^b)$, namely bitwise XORs, and $\bigoplus$ as the accumulated sum operator. $\|$ represents concatenation of two vectors.

5.1.2. Elements in data packets

The information part carrying the health data from sensors or commands from controllers is denoted by k, and r is the ECC redundancy to protect k from random errors. $y = k||r$ is the concatenation of both. The self-increment timestamp is denoted by i, and the random vector by x. The Robust code's signature is denoted by ω_{Rob}, and the AMD code's signature by ω_{AMD}.

5.1.3. Attacks

e represents the injected error by attackers to each data packet and so $e = \{e_\omega, e_y, e_i, e_x\}$. Any packet tampered by e is marked by $\tilde{}$. The attack mis-detection probability is denoted by P_{miss}.

5.1.4. Random error correction

The ECC's check matrix H is used with $\tilde{y}$ to compute the syndrome S for random error correction.

5.2. Robust codes against weak attacks

Robust codes are often used in cryptosystems for their high security attribution (Tomashevich et al., 2014). They generate a non-linear signature of a message for authentication. Robust codes are designed based on the assumption that the attackers cannot predict the content of a message, which falls into our definition of the weak attack.

Definition 5.1. Denoting a message as v and the digits in this message as: v_i where $v = (v_0, v_1, \cdots, v_{N-1})$, $v_i \in GF(2^b)$, the Robust code's signature is calculated as follows (Neumeier and Keren, 2012):

$$\omega_{Rob} = \begin{cases} \bigoplus_{i=0}^{(N-2)/2}(v_{2i} \cdot v_{2i+1}), & N \text{ is even;} \\ v_{N-1}^3 \oplus \left[\bigoplus_{i=0}^{(N-3)/2}(v_{2i} \cdot v_{2i+1})\right], & N \text{ is odd.} \end{cases}$$

For the IMD case, we consider $v = y||i = k||r||i$. Therefore the equation above becomes:

$$\omega_{Rob} = y \cdot i. \tag{1}$$

The Robust decoder verifies if the following equation holds:

$$\tilde{\omega_{Rob}} \stackrel{?}{=} (\tilde{k}||\tilde{r}) \cdot \tilde{i}. \tag{2}$$

If the injected errors to each component are represented as e_ω, e_y, and e_i, the error masking equation will be:

$$\omega_{Rob} \oplus e_\omega = [(k||r) \oplus e_y] \cdot (i \oplus e_i). \tag{3}$$

It has been proven that the right-hand side of the equation is always a non-zero polynomial of i or y of degree 1. It is fairly straightforward to also prove that for a certain message and an error e, the probability of missing this error e is at most.

$$P_{miss} = 2^{-b}. \tag{4}$$

According to the design criteria, b should be at least 32 or 40 to ensure that no attack will succeed in an IMD's lifetime.

5.3. AMD codes

The AMD codes have been known as a class of lightweight but highly secure attack detecting codes that are effective against

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

strong attacks. They often used in conjunction with cryptographic systems as a keyless authentication code (Cramer et al., 2013). Because of its random vector x, AMD codes performs well with uniform security even under non-uniform distribution of the information part, which covers the vulnerability of the highly repetitive health data or commands in IMDs. In strong attacks we assume that attackers have knowledge of the information part, the encoding scheme, and are able to issue any modifications to the message in the channel.

Definition 5.2. Let $x = (x_1, x_2, \cdots, x_m)$, where $R_i \in GF(2^b)$ is a randomly generated b-bit vector. An h^{th} order ($h \leq 2^b - 2$) Generalized Reed-Muller code (GRM) with m variables consists of all codewords $(f(0), f(1), \cdots, f(2^{bm}-1))$, where $f(x)$ is a polynomial of $x = (x_1, x_2, \cdots, x_m)$ of degree up to h. Let

$$A(x) = \begin{cases} \bigoplus_{i=1}^{m} x_i^{h+2}, & \text{if } h \text{ is odd;} \\ \bigoplus_{i=2}^{m-1} x_1 x_i^{h+1}, & \text{if } h \text{ is even and } m > 1; \end{cases}$$

where $\bigoplus$ is the accumulated sum in $GF(2^b)$. Let

$$B(x, y) = \bigoplus_{1 \leq j_1 + j_2 + \cdots + j_1 \leq h+1} y_{j_1, j_2, \cdots, j_m} \prod_{i=1}^{m} x_i^{j_i},$$

where $\prod_{i=1}^{m} x_i^{j_i}$ is a monomial of x of a degree between 1 and $h+1$. And $\prod_{i=1}^{m} x_i^{j_i} \notin \triangle B(x, y)$ which is defined by:

$$\begin{cases} \{x_1^{h+1}, x_2^{h+1}, \cdots, x_m^{h+1}\}, \text{if } h \text{ is odd;} \\ \{x_2^{h+1}, x_1 x_2^{h}, \cdots, x_1 x_m^{h}\}, \text{if } h \text{ is even and } m > 1. \end{cases}$$

Let $f(x, y) = A(x) \oplus B(x, y)$, then a generalized AMD codeword is composed of the vectors $(y, x, f(x, y))$, where y is the information portion, x the random vector, and $f(x, y)$ the redundancy portion (Wang and Karpovsky, 2011).

Remark 5.1. If the attack involves an error $e_y \neq 0$ on the information y, which is the major purpose of almost all attacks, then in $f(x, y)$ the term $A(x)$ can be omitted.

For the proposed protection scheme, $m = 1$ since $y = k||r$ is in one packet. y can be robustly combined with the self-incrementing timestamp i by $y \cdot i$, where $\cdot$ is the finite field multiplication. The signature ω of the AMD code is computed as:

$$\omega_{\text{AMD}} = y \cdot i \cdot x = (k||r) \cdot i \cdot x. \tag{5}$$

The AMD decoder verifies if the following equation holds:

$$\omega_{\tilde{\text{AMD}}} \stackrel{?}{=} (\tilde{k}||\tilde{r}) \cdot \tilde{i} \cdot \tilde{x}. \tag{6}$$

If the injected errors to each component are represented as e_ω, e_y, e_i and e_x, the error masking equation will be:

$$\omega_{\text{AMD}} \oplus e_\omega = [(k||r) \oplus e_y] \cdot (i \oplus e_i) \cdot (x \oplus e_x). \tag{7}$$

It has been shown that the right-hand side of the equation is always a non-zero polynomial of x of degree 1. Similar to equation 4, it can be shown that the probability of missing this error e is at most

$$P_{miss} = 2^{-b}, \tag{8}$$

for a given a message and an error, b should be at least 32 or 40 to ensure that no attack will succeed in an IMD's lifetime.

5.4. *Error correction codes for random errors*

Like in most electronic devices, some basic error correcting code's (ECC) redundancy is added to information part of the IMD sensor or controller message to help recover from random errors. Since the proposed scheme uses at least 32-bit packets and the information part is at most 16 bits, the rest of the bits can be allocated for the ECC's redundancy.

To ensure fast decoding and low hardware complexity, we propose to use the Orthogonal Latin Square Codes (OLSCs) (Yalcin et al., 2014). The error correction procedure is:

$$H \cdot (\tilde{k}||\tilde{r}) = S \tag{9}$$

where $\tilde{k} \in GF(2^{16})$ and $\tilde{r} \in GF(2^{16})$ are the distorted information part and redundancy, H is a 16×32 binary matrix, and S is a 16-bit binary vector which is used for one-step majority voting error correction of up to 2 random errors in k.

5.5. *System diagram*

As stated in the previous section, the proposed scheme is structured as an authenticated encryption with a MAC-then-Encrypt workflow. The AES-CBC encryption process will protect the system from eavesdropping on k, the health data or commands. The ECC's redundancy r enables correction of up to 2 random errors in k. The timestamp i will guarantee that each transmitted cipher cannot be replayed again to spoof a legal command or health data. The random vector x randomizes the plaintext $((k||r)||i||x||\omega)$ against strong attacks (for weak attacks x will be set to 1). The Robust or AMD authenticating signature ω verifies if the message is authentic or not. The system diagram is shown in Fig. 6.

Protocol 5.1. The encoding and decoding procedure of the system is as follows:

1. On the sender's side, the system first encodes the health data k from the sensors or commands from the controller into the 32-bit information part y, with a double-error correcting (ECC) OLSC code redundancy r;
2. The 32-bit information part is then encoded with a timestamp i and random vector x into ω by the Robust or AMD message authentication code;
3. The $((k||r)||i||x||\omega)$ will serve as the plaintext to be encrypted by the AES module before being sent to the wireless channel;
4. On the receiver end, the AES module will decrypt the cipher;
5. The first IMD information and its redundancy are sent to the OLSC decoder for random error correction;
6. i is compared with the previous timestamp to check if it is up to date or a replayed message;

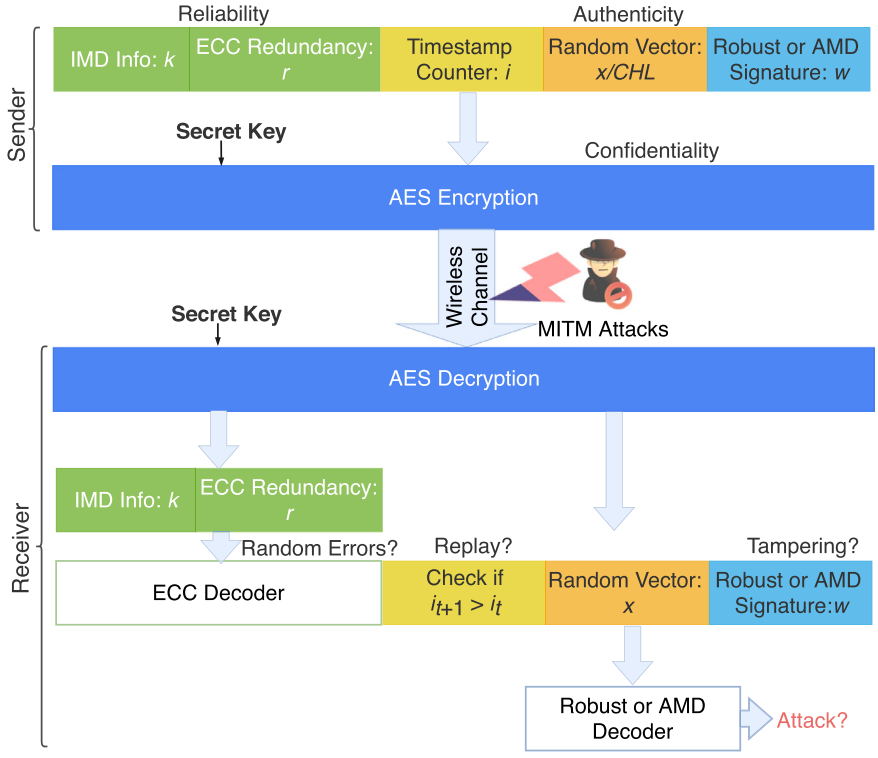

Fig. 6 – The entire system works under data blocks of size 128-bit with sub-block size 32-bit, or 192-bit block size with each sub-block in 48 bits. Such a block encapsulates authentication, obfuscation, timestamp, and random error correction to provide both reliability and security to the transmission channel.

7. Finally, the entire plaintext is verified by the Robust or AMD decoder to detect any tampering attacks.

Remark 5.2. Since x serves as one of the most important variables in obfuscation and authentication, its confidentiality needs to be guaranteed. Although it is protected by the encryption module, we also consider the most severe scenario where the secret key of MtE is leaked, thus the attackers will be able to read the cipher as plaintext. In this case there needs to be a zero-knowledge approach in transferring x to the decoder end.

Since all IMDs are hardware devices, we take advantage of the physical unclonable functions (PUFs). A PUF (Gassend et al., 2002) is a piece of hardware that produces unpredictable responses upon challenges due to their manufacturing variations. PUFs work on the challenge and response pairs (CRPs). Each PUF's output (response RSP_i) is a non-linear function of the outside stimulation (challenge CHL_i) and the PUF's own physical, intrinsic, and unique diversity. Therefore, even under exactly the same circuit layout and manufacturing procedure, two PUF-based IMD pieces will give distinct responses under the same challenge. PUFs are mostly used to verify the authenticity of a hardware device. However, in this paper we use it for secret variable updating. Its working principle is as follows:

(i) When a new IMD is produced, the manufacturer uses $\{CHL_0, CHL_1, \cdots, CHL_j, \cdots CHL_n\}$ as the challenges to the PUF in the IMD. Then the corresponding responses $\{x_0, x_1, \cdots, x_j, \cdots x_n\}$ are stored in the IMD controller/configuration device. The challenge and response pairs (CRPs) have to be acquired this way since even the manufacturer cannot predict the CRP values before they are produced;

(ii) After the device is deployed, when the controller/configuration module needs to send a message, e.g., an insulin injection command, it must use one of the responses, e.g., x_j, as the random variable for the AMD encoding. To inform the IMD's AMD decoder of the choice, it uses CHL_j instead of x_j in the encoded data block of Fig. 6;

(iii) The IMD applies CHL_j to its PUF to locally generate the associated response x_j. It can then use this x_j as the AMD random variable to decode the message and execute the command.

The above procedure is secure since CHL_j leaks zero knowledge of x_j. An attacker cannot learn x_j in order to forge legal command codes even if they acquire the MtE key. With this approach, if the authentication of the transmitted messages is the only protection on the device - this is the case in most existing attacks (error injection, forged command codes, replay etc.) – then the PUF with AMD codes is sufficient to secure the IMD. Therefore, even if the AES module is not enabled, forging a legal AMD signature is not feasible.

Another advantage for using a PUF-based communication approach is that it uniquely links an IMD to a controller/configuration device. Other controllers, either legally

manufactured for other IMDs or illegally forged for attack purposes, will not be able to establish a verifiable communication with this IMD, since they do not have the corresponding CRP stored.

6. Third party authorization

There may be a need for a legitimate third party to interfere at the time of emergency. While the wireless transmission channel of an IMD is secured against attackers, it has to grant access upon the requests from medical teams. During an emergency, the patient or user may have lost consciousness and may be not able to provide any information about the IMD. Thus the medical team needs to acquire the necessary information from the manufacturer or service point of the IMD by themselves. The procedure steps are: 1) the medical team makes a request to the IMD service point; 2) the access of the IMD is granted to the team after the verification of the request, as shown in the Fig. 7.

6.1. *Threshold authorization protocol*

For security reasons, there need to be restrictions on the authorization of the secret key. It should not be entrusted to all individuals who request it. Otherwise, if an attacker is able to acquire the legal credential of a medical worker, they would be able to gain full control of the targeted IMD. Ideally, the secret key should be granted based on different levels of trustworthiness of medical staff. For example, a surgeon or physician alone should be able to request the access key, while it takes at least one emergency medical technician (EMT) and one paramedic together in an ambulance to make the request.

Protocol 6.1. For the third party medical team to be authorized with the secret key of an IMD, the threshold authentication protocol is as follows:

1. The IMD or its corresponding medical system decides on a threshold t. Only when the total request level reaches t, can the secret key or other access credentials be authorized;
2. Each medical worker is given one or multiple request tokens based on their function and authorized participation levels;
3. When a medical worker request access, if the sum of their request levels reaches t, the access to the IMD is granted. Below t, the IMD is unaccessible.

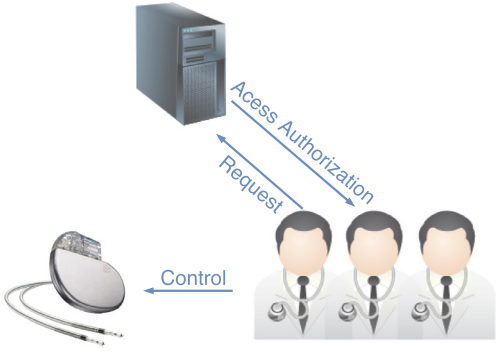

Fig. 7 – The access can involve the secret key or even the CRPs as mentioned in Remark 5.2.

It is worth noting that the demand above fits naturally to the property of Shamir's threshold secret sharing, which will be used to implement the protocol.

6.2. *Threshold secret sharing*

The following notations are introduced first:

- A: the authorization request tag;
- D_i: the public ID of the ith medical worker;
- h_i: the request token held by the ith medical worker;
- t: the threshold of the request;
- $\oplus$: the addition operator in finite fields;
- $\cdot$: the multiplication operator in finite fields;
- $\bigoplus$: the cumulative sum operator in finite fields;
- $\prod$: the cumulative product operator in finite fields.

The concept of t-threshold secret sharing (TSS) was first introduced by Shamir (1979) and studied by many researchers (Bu et al., 2017; Pang and Wang, 2005). All the computations are carried out over Galois finite field (GF) arithmetic. To share a secret A, a polynomial of degree $(t-1)$ is used to compute and distribute the shares, where the secret A serves as the free or leading coefficient, and all other coefficients can be arbitrarily chosen. The shares are the evaluations of the polynomial by each holder's D_i. The share distribution equation is

$$h_i = a_0 \oplus a_1 D_i \oplus a_2 D_i^2 \oplus \cdots \oplus A D_i^{t-1}. \tag{10}$$

where $A, h_i, D_i \in GF(2^b)$ and b is the length of these vectors.

The ID number D_i is publicly known to everyone while the shares h_i are kept private by each shareholder.

With any subset of at least t shareholders' IDs and shares, one can use the Lagrange interpolation formula to reconstruct the secret:

$$A = \bigoplus_{i=0}^{t-1} \frac{h_i}{\prod_{j=0, j\neq i}^{t-1} (D_i \oplus D_j)}. \tag{11}$$

Such a construction is $(t-1)$-private. This means it needs at least t shareholders to reconstruct the secret and so any $(t-1)$ or less shareholders have zero knowledge of the secret.

In the language of the threshold authorization of access for IMDs, the secret A is the authorization request tag securely stored at the IMD service point. The share h_i is the request token computed based on A and the medical workers' IDs D_i, and then distributed to them. The tag A can only be constructed by t (or more) request tokens from medical workers. Once it is submitted to the service point and verified successfully, the access (secret keys or CRPs of the PUF) to the IMD will be granted. To any medical team with less than t request tokens, A is kept information theoretically private.

The distribution of the request tokens h_i can be leveraged to provide different medical workers with different levels of request privileges, based on their occupation or trustworthiness. Fig. 8 illustrates a possible application scenario.

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

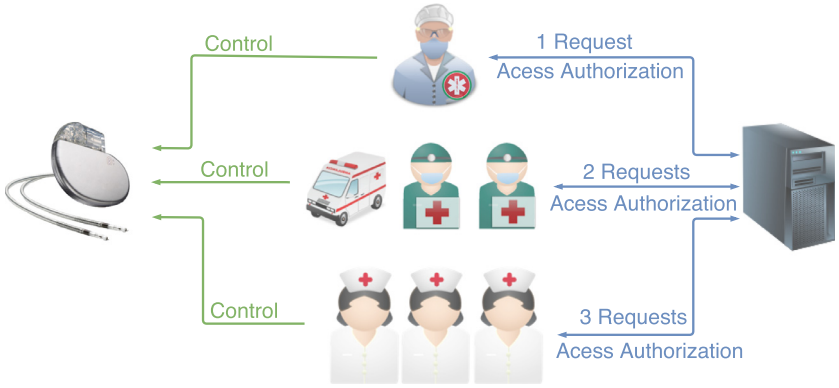

Fig. 8 – For example, a hospital can set the request threshold to $t = 3$ and allow surgeons or physicians to be entrusted with 3 request tokens while each EMT or paramedic on an ambulance is entrusted with 2 tokens and a nurse with 1. In case of an emergency, it will need at least 3 nurses to request for the access to the IMD, or 2 ambulance workers, or one surgeon or physician.

Thus, the demand of Protocol 6.1 is met, and the security issue mentioned in Section 6.1 is resolved.

Example 6.1. In this toy example we show how the threshold authorization protocol functions with $t = 3$.

In the Longwood Hospital, the secret authorization request tag is $A \in GF(2^{16})$ where $A = 0x3F01$. It is shared in a zero-knowledge manner to all the 7 hospital staff with a threshold $t = 3$ and the following distribution equation:

$$h_i = a_0 \oplus a_1 D_i \oplus A D_i^2$$

where $a_0 = 0xAAAA$, $a_1 = 0x5555$ are arbitrarily chosen coefficients and $a_0, a_1, A \in GF(2^{16})$. This secret tag is shared to seven shareholders with IDs and shares:

$$\{D_1 : h_1\} = \{0x0001 : 0xC0FE\}$$
$$\{D_2 : h_2\} = \{0x0002 : 0xFC04\}$$
$$\{D_3 : h_3\} = \{0x0003 : 0x9650\}$$
$$\{D_4 : h_4\} = \{0x0004 : 0x0FB4\}$$
$$\{D_5 : h_5\} = \{0x0005 : 0x65E0\}$$
$$\{D_6 : h_6\} = \{0x0006 : 0x591A\}$$
$$\{D_7 : h_7\} = \{0x0007 : 0x334E\}$$

In an emergency, the staff members with IDs {0x0003, 0x0005, 0x0007} were sent to a patient. Since $t = 3$, by substituting their IDs and shares into [Eq. 11], they are able to reconstruct $A = 0x3F01$ and legally configure the patient's IMD. still cannot break into the patient's IMD

An attacker cannot break into the patient's IMD even if they manage to compromise at most two staff members, e.g., {0x0004, 0x0006}. Since any number of staff less than t, [Eq. 11] leaks no knowledge of A.

7. Evaluations

In this section the proposed scheme's security and power consumption overhead will be evaluated.

7.1. *Error mis-detection probabilities*

To validate the mis-detection probability we run tests on over three billion simulated IMD radio transmissions on sensor's health data and controller's commands. This is estimated the total number of an IMD's transmissions in 10 years under a frequency of 10Hz. During the simulation, the system mimics an IMD sending and receiving messages, while the attacker applying hijack, replay, and bit-flipping attacks alternatively in every round, and the receiver verifies the timestamps and the Robust or AMD signatures.

As stated in Section 5.5, with the 128- or 192-bit AES modules in most IMDs, we are able to apply 32- or 48-bit AMD code signatures for authentication. In our experiments, the 32-bit and 48-bit systems provided strong security where not a single attack was successful among over three billion transmissions/attacks. We also applied various sizes of sub-blocks (from 8 to 48 bits) due to AMD codes' flexibility to observe how much the experimental error mis-detection probability matches $P_{miss} = 2^{-b}$ in (Eq. (4)), as shown in Table 1.

The experimental results not only show that the proposed protection scheme works well according to the theoretical estimation of 2^{-b} error mis-detection probability, but also show that the 32-bit and 48-bit schemes are secure enough to miss no attack in over three billion mounted attacks. Essentially, the proposed design technique provides sufficient security for the lifespan of an IMD.

7.2. *Power consumption overhead*

As mentioned above, Robust and AMD codes are lightweight message authentication codes. With the AES enabled in the IMDs, the encoding and authentication add little/small power consumption overhead while providing the security demanded. This is critical for power sensitive IMDs such as defibrillators whose battery replacement requires surgery.

The overhead comparison in Table 2 was made based on the implementation on a Xilinx Virtex 4 FPGA and Cadence SOC Encounter. The communication channel is constructed

Table 1 – P_{miss} under 3,154,043,200 Active MITM Attacks.

b				
Missing	8	16	32	48
Missed Errors in Experiments	12,321,649	48,032	0	0
Experimental P_{miss}	3.91e−3	1.52e−5	0	0
Theoretical $P_{miss} = 2^{-b}$	3.91e−3	1.53e−5	2.33e−10	3.55e−15

1 Under 3 billions MITM attacks modeled in Section 3, not a single error was mis-detected by the 32-bit and 48-bit packet-sized systems.

Table 2 – Power Overhead Comparison Based on AES enabled.

	P_{miss}	Extra bits Over AES	Area (um^2)	Area Overhead	Energy (nJ)	Energy Overhead
Proposed Scheme (32-bit packets)	2^{-32}	0	3093.6	5.37%	2.10	3.13%
AES (128 bits)	N/A	N/A	57520.3	N/A	67.03	N/A
Proposed Scheme (48-bit packets)	2^{-48}	0	4765.9	7.14%	4.05	4.43%
AES (192 bits)	N/A	N/A	66732.7	N/A	91.36	N/A

1 The proposed authentication module adds only 3.1% energy to the 128-bit AES encryption module, and 4.4% to the 192-bit AES module, resulting in an ignorable energy consumption overhead while providing sufficient security.

Table 3 – Transmission and power overhead comparison.

	P_{miss}	Extra bits Over AES	Area (um^2)	Energy (nJ)
Proposed Scheme (32-bit packets)	2^{-32}	0	3093.6	2.10
Proposed Scheme (48-bit packets)	2^{-48}	0	4765.9	4.05
Proposed Scheme (80-bit packets)	2^{-80}	128	6274.8	7.49
HMAC Based (160 bits)	2^{-80}	128	58813.7	58.06

1 Even when the authentication process is brought up to error mis-detection probability of 2^{-80} which is the same as HMAC, the hardware and energy costs are only 10.7% and 12.9% of the later, making the proposed lightweight scheme an economic choice for the IMDs.

with the same parameters of the IMDs where 128- or 192-bit AES-CBC is adopted as the encryption mode, and the plaintexts (sensor measurements and commands) are not larger than four bytes.

On another hand, one alternative approach is AES + HMAC + timestamps. However, HMAC requires at least 160 bits to provide 2^{-80} mis-detection probability which is unnecessary for the security required and involves a significant amount of modifications to the existing AES based systems.

As for the 32-bit and 48-bit Robust or AMD code and timestamp based scheme, since all computations are done in the 32-bit or 48-bit finite field, there are less overall transmission overhead, hardware area usage and power consumption, compared to the HMAC authentication method. Even if the scheme upgrades x and ω to 80 bits to achieve the same P_{miss} as the HMAC based scheme, it will still save a significant amount of system power consumption as shown in Table 3.

8. Conclusion

In this work we propose a technique to address the existing and potential Man-In-The-Middle attacks on implantable medical devices' wireless communication. We prove theoretically and through experimental results that by using authenticated encryption with a random vector and a timestamp encoded by Robust or AMD codes. The proposed design misdetects no error in the simulated device's lifespan. Depending on the attack model, different authentication approaches can be used to achieve cost-efficiency. Robust codes with less hardware complexity could be used if the attackers have minimal knowledge of the transmitted packet contents, while AMD codes with a higher hardware cost could be used to protect against more knowledgeable attackers. Moreover, the proposed authentication module's energy consumption is a mere

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

3–4% of the pre-installed AES module. Compared with other authentication techniques such as HMAC, our approach consumes only 13% of their energy while providing the same security level. These advantages make the proposed scheme a secure and reliable solution for IMDs to defend against MITM attacks, while extending the lifespan of IMDs by preserving battery life.

In addition, we also propose a third party authorization protocol. In case of a medical emergency, the patient may not be able to provide any information about the IMD device. Therefore, there needs to be a provably secure access mechanism in place for the third party medical team to service the device. We design a threshold-based authorization protocol, which takes a number of medical workers to request access to IMDs based on their trust levels. Therefore, the access to the IMD is managed in a secure and robust manner.

Acknowledgments

This research is partially supported by the NSF grants (No. CNS-1745808) and (No. CNS-1012910).

REFERENCES

Bu L, Karpovsky M. A hybrid self-diagnosis mechanism with defective nodes locating and attack detection for parallel computing systems. On-Line Testing and Robust System Design (IOLTS), 2016 IEEE 22nd International Symposium on 2016.

Bu L, Karpovsky MG. In: A design of secure and reliablewireless transmission channel for implantable medical devices.; 2017. p. 233–42.

Bu L, Nguyen HD, Kinsy MA. Rasss: A perfidy-aware protocol for designing trustworthy distributed systems. Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2017 IEEE International Symposium on 2017:1–6.

Burleson W, Clark SS, Ransford B, Fu K. Design challenges for secure implantable medical devices. Springer New York 2014.

Chede S, Kula K. Design overview of processor based implantable pacemaker. Journal of Computers 2008;3(8):49–57.

Cramer R, Fehr S, Padró C. Algebraic manipulation detection codes. Science China Mathematics 2013;56(7):1349–58.

Daemen J, Rijmen V. The design of rijndael: Aes-the advanced encryption standard. Springer Science and Business Media 2013.

Gassend B, et al. Silicon physical random functions. Proceedings of the Computer and Communications Security Conference 2002.

Halperin D, Heydt-Benjamin TS, Ransford B, Clark SS, Defend B, Morgan W, Fu K, Kohno T, Maisel WH. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. Proceedings of the 29th IEEE Symposium on Security and Privacy 2008.

InfoSec. Hacking implantable medical devices, 2014. http://resources.infosecinstitute.com/hcking-implantable-medical-devices/

Krawczyk H. The order of encryption and authentication for protecting communications (or: How secure is ssl?). Springer Berlin Heidelberg 2001.

Li C, Raghunathan A, Jha NK. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. Proceedings of the 13th IEEE International Conference on e-Health Networking, Applications, and Services 2011a.

Li C, Raghunathan A, Jha NK. In: Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. IEEE; 2011b. p. 150–6.

McDonald J, Dean S, Niewolny D, Garcia D, Chhabra N, Chang L. Integrated circuits for implantable medical devices. Freescale Solutions for the Medical Market, 2011.

MedWOW, 2018. http://www.medwow.com.

Neumeier Y, Keren O. Punctured karpovsky-taubin binary robust error detecting codes for cryptographic devices. In: On-Line Testing Symposium (IOLTS), 2012 IEEE 18th International. IEEE; 2012. p. 156–61.

Pang LJ, Wang YM. A new (t, n) multi-secret sharing scheme based on shamirs secret sharing. Applied Mathematics and Computation 2005;167(2):840–8.

Paul N, Kohno T, Klonoff DC. A review of the security of insulin pump infusion systems. Journal of Diabetes Science and Technology 2011;5(6):1557–62.

Radcliffe J. Hacking medical devices for fun and insulin: Breaking the human scada system. Black Hat Conference 2011.

Roberts P. Blind attack on wireless insulin pumps could deliver lethal dose, 2011. http://threatpost.com/en_us/blogs/blind-attack-wirelessinsulin-pumps-could-deliver-lethal-dose-102711.

Rostami M, Burleson W, Koushanfar F, Juels A. Balancing security and utility in medical devices? Proceedings of the 50th Annual Design Automation Conference 2013.

Rushanan M, Rubin AD, Kune DF, Swanson CM. Sok: Security and privacy in implantable medical devices and body area networks. 2014 IEEE Symposium on Security and Privacy 2014.

Shamir A. How to share a secret. Communications of the ACM 1979;22(11):612–13.

Swepsie. Bypassing encrypted session tokens using cbc bit flipping technique, 2014. http://swepssecurity.blogspot.com/2014/05/bypassing-encrypted-session-tokens.html.

Takahashi D. Excuse me while i turn off your insulin pump, 2011. http://venturebeat.com/2011/08/04/excuse-mewhile-i-turn-off-your-insulin-pump/.

Tomashevich V, Neumeier Y, Kumar R, Keren O, Polian I. Protecting cryptographic hardware against malicious attacks by nonlinear robust codes. In: Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on. IEEE; 2014. p. 40–5.

Wang Z, Karpovsky MG. Manipulation detection codes and their application for design of secure cryptographic devices. Proc of International Symposium on On-Line Testing (IOLTS) 2011.

Yalcin G, Islek E, Tozlu O, Reviriego P, Cristal A, Unsal OS, Ergin O. Exploiting a fast and simple ecc for scaling supply voltage in level-1 caches. IEEE International On-Line Testing Symposium (IOLTS) 2014.

Yury C. Your heartbeat may soon be your only password, 2014. http://wired.com/insights/2014/06/heartbeat-may-soon-password/.

Lake Bu is a PhD candidate in the Adaptive and Secure Computing Lab at Boston University. His research focus is on hardware reliability and security. His goal is to design reliable, secure, and robust hardware systems for distributed systems, whose communications' confidentiality and authenticity can be ensured. He studies various types of threats on hardware systems such as random errors, error injection attacks, Man-In-The-Middle attacks, and collusive attacks. He explores defense approaches using security oriented codes, secure secret sharing schemes, and cryptography etc. on FPGA platforms.

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011

Mark G. Karpovsky has been Professor of Electrical and Computer Engineering at Boston University, and Director of the Reliable Computing Laboratory. He conducts research in the areas of new techniques for design of reliable multiprocessors, networks of workstations and local area networks, routing in computer and communications networks, testing, and diagnosis of computer networks combining on-line and off-line techniques for error detection and/or location, and fault-tolerant message routing for computer networks.

Michel A. Kinsy is an Assistant Professor in the Department of Electrical and Computer Engineering at Boston University (BU), where he directs the Adaptive and Secure Computing Systems (ASCS) Laboratory. He focuses his research on computer architecture, in particular, secure architecture designs, hardware-level security, neural network accelerator designs, fault-tolerant network-on-chip routing algorithms, and cyber-physical systems. Dr. Kinsy earned his PhD in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 2013.

Please cite this article as: Lake Bu, Mark G. Karpovsky and Michel A. Kinsy, Bulwark: Securing implantable medical devices communication channels, Computers & Security, https://doi.org/10.1016/j.cose.2018.10.011