Bulwark: Securing Implantable Medical Devices Communication Channels

Lake Bu†[1], Mark G. Karpovsky§, Michel A. Kinsy†

†Adaptive and Secure Computing Systems Laboratory
§ Reliable Computing Laboratory
Department of Electrical and Computer Engineering,
Boston University, Boston, USA

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 balanced security, reliability, and power consumption. Therefore, in this work, we propose a robust approach for guarding against existing and potential communication-based attacks in 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.

Keywords: implantable medical devices; security; man-in-the-middle; encryption; authentication; Robust codes; AMD codes.

1. Introduction

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

In the past decade, various types of attacks to the IMDs have been reported. According to the researches, many have been successful in either acquiring the health data or manipulating the functionality of the IMDs. Many attacks can be applied wirelessly which are categorized as Man-In-The-Middle (MITM). To add insult to injury, 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 not too difficult to predict the information even if it is encrypted, which makes the IMDs more vulnerable to attacks.

Eavesdropping is one of the most commonly seen passive attacks to wireless channels. The attackers simply listen to the unencrypted transmissions and acquire the knowledge of the health of the targeted patients or victims. Since there is no malicious tampering to the transmission, it is hard to detect. There are software and hardware means to eavesdrop the IMDs' channel. Researches on this type of passive attacks have been made by [Halperin et al. (2008)], [Li et al. (2011a)], and [Paul et al. (2011)] etc.

Whereas eavesdropping only aims at stealing the victim's medical information, active attacks such as hijacking or replay are more lethal 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 explored by [Halperin et al. (2008)] over pacemakers and [Roberts (2011)] over insulin pumps. These attacks have been shown in simulation to result in fatal attacks.

Although, some IMD manufacturers have integrated the Advanced Encryption Standard (AES) in their devices, they are often not activated due to power consumption concerns or authentication complexity of the third party devices [InfoSec (2014)]. Moreover, even if the AES module is activated and the transmission is encrypted, they are properly authenticated. These functionality-oriented choices make it possible for attackers to maliciously exploit the devices and thus harm their users.

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 packets should be protected;
- When all the security features are turned on, the power consumption overhead should be negligible (below 10%);

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

- The third party (emergency team or hospital) authentication should achieve convenience and security at the same time.

Therefore, in this paper we extended our previous work [Bu and Karpovsky (2017)] and developed an 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 is added with small hardware and power overhead (less than 10%);
- It provides a threshold-based third-party authorization protocol for the time of a medical emergency, ensuring availability of the medical service with 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 introduces a protocol of third party authentication during emergency. Section 7 evaluates the proposed design by experiments and overhead comparison with other possible 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. They receive wireless transmissions from the sensor inside the patients' bodies 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)].

Researches have shown that many IMDs in this type have the Advanced Encryption Standard (AES) modules built in, they are generally not turned on due to the power issue. Some of these devices are: Medtronic Maximo implantable cardiac defibrillator (ICD) series, BIOTRONIK Itrevia 7 DR-T/VR-T ICD series [Rostami et al. (2013)].

Since the transmissions in such an IMD are generally not encrypted, the messages are plainly transferred from the medical sensors. Thus it is not too difficult to eavesdrop and learn patients' information. Based on the acquired information, the

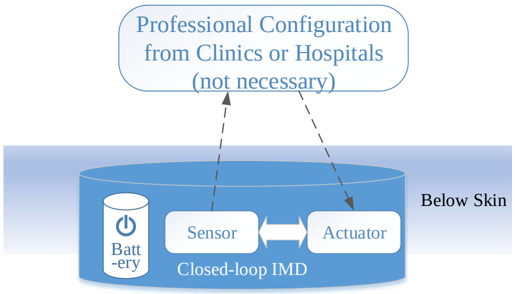

Figure 1: Closed-loop IMDs manage themselves based on the communication between the sensor and the actuator. Although they have no access for the patients to control them, they do allow configurations from professionals. The communication is generally not encrypted due to the power consumption consideration.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

attackers will be able to replay some of the messages to the monitor, inducing the device to react in a certain way. In 2008, [Halperin et al. (2008)] conducted their research on the vulnerabilities of pacemakers and ICDs. They successfully listened to and understood the wirelessly transmitted information of the patient. Furthermore, they were even able to reuse the stored messages to disable the device, which may cause fatal accidents.

Besides, the power consumption is another major issue in these IMDs. Usually pacemakers or ICDs are designed to last for 5 to 7 years. Once the battery runs out of power, it takes a surgery 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. They receive wireless transmissions from the devices' sensors inside patients, who are able to respond with remote controls. According to the reading of his/her glucose, the patient 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 makes even simple attacks to have particularly harmful impacts on the victims.

[Li et al. (2011a)] have also studied the case on insulin pump systems. They were not only able to acquire the encrypted information from the device, but also managed to forge false glucose readings to the monitor. Finally, they successfully sent their own commands to the pump due to its lack of authentication. Moreover, other researchers such as [Radcliffe (2011)] and [Takahashi (2011)] claimed that they have gained full con-

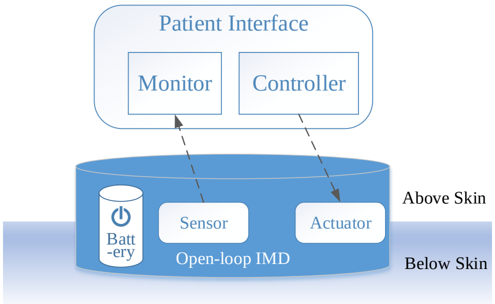

Figure 2: Open-loop IMDs usually come with a monitor and a controller. The patients monitor their health status based on the data from the sensor. They can then issue commands (such as an exact dosage of medicine or starting the insulin pump) based on the information provided. The communication is not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

trol to some of the insulin pump systems, since these devices accept unauthorized radio signals or commands.

2.3. Biosensors

Biosensors are different from the two types above in several ways. Firstly they are usually self-powered inside human bodies. Secondly they are purely transmitters and receive no commands. Biosensors are widely used to detect glucose, lactate, or cholesterol etc. The receiver (patches) serves as the middle station which powers the sensor while sending the data to a higher level of monitors or analysts. However, both the monitor and the patch give no feedback in the form of commands.

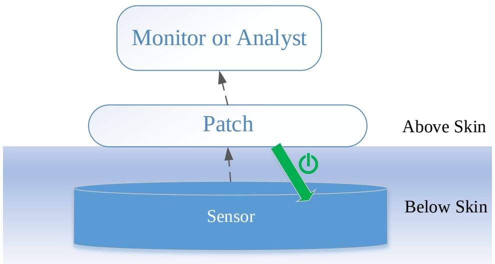

Figure 3: Biosensors send measurement to the patch and are powered by it. The patch then sends the measurement to monitors for analysis.

The major threat to biosensors will be eavesdropping although it is not easy to implement because of their short communication distance. Other precise and practical threat models are yet to be developed [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. Also many IMDs are equipped with AES but have not enabled the encryption modules (c.f. Section 2). Even if they have, this can only prevent the attackers from eavesdropping and understanding the patients' health information. Those devices may still be vulnerable in certain ways under active Man-In-The-Middle (MITM) attacks such as hijack and replay. We make the following definitions on weak and strong attack models.

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 to the victim's actuator [Rushanan et al. (2014)];

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. The attacker can eavesdrop the victim's wireless IMD transmission between the sensor and the monitor/controller [Rostami et al. (2013)];

2. The attacker is able to use a programmed radio to interfere with the transmitted packets to the victim's actuator [Rushanan et al. (2014)];

3. The attacker has the ability to acquire the format of the data packets, and make a reasonable estimation or prediction [Yury (2014), Rushanan et al. (2014)] of the information they carry.

For the strong attack, learning the IMD data packet format is not easy, but possible. First, usually the IMD manufacturers do not publish the message format or the command codes. However, with enough effort they can be searched out. Without official instructions, it does take some effort to figure out this information, since usually one IMD can have multiple distinct packet formats.

Secondly, to learn about a transmitted data packet in real time, may require wireless peripherals such as the Texas Instruments (TI) CC1101 RF Transceiver Modules, in order for very low-power wireless applications to capture and buffer the packets for analyses. In this case, the transmission frequency also needs to be calculated [Radcliffe (2011)]. For most IMDs, the transmission frequency is fairly low, *e.g.*, one in five minutes for insulin pumps.

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

3.1. IMDs with AES Disabled

As mentioned in Section 2, most IMDs have their 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. The results can be 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 AES Enabled

Even if the IMDs activate their AES module to have each transmitted message encrypted, there still can be many potential ways to attack them [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 the 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 authentication of the incoming radio signals. Thus the attackers and establish anonymous transmissions to either the implanted device or the monitor/controller.

This gives attackers an opportunity to take over the transmissions between legal sensors and controllers. The attacker can first eavesdrop and record the legal transmitted ciphers without any understanding of the health data. Then he/she can replay some of the stored legal ciphers to IMD. Even if every transmitted message is encrypted and authenticated, the replayed ciphers will still be considered as legal. Moreover, if the attacker is able to apply eavesdropping and replay attacks, he/she can choose to inject certain ciphers to harmfully affect the IMD. For example, very high glucose measurements can be frequently sent to the patient's monitor, inducing overdose of insulin pumps [Rushanan et al. (2014)] . Commands of persistent large electric shocks can be sent to the defibrillator or pacemaker, causing a deadly health event, *e.g.*, a cardiac arrest.

For example, if an IMD has AES with Cipher Block Chaining (CBC) mode turned on, the current plaintext will be randomized by the previous cipher and then sent for encryption. Thus the decrypted texts are beyond the control of the attackers. However, since the health data are usually generated from microprocessors and sensors of 8 bits, 12 bits, or 16 bits [Chede and Kula (2008), McDonald et al. (2011)], it makes the replayed cipher from attackers decrypted to another legal numeric value with a high probability.

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 he/she has 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 attacker 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 level glucose in the blood 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 he/she has 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 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

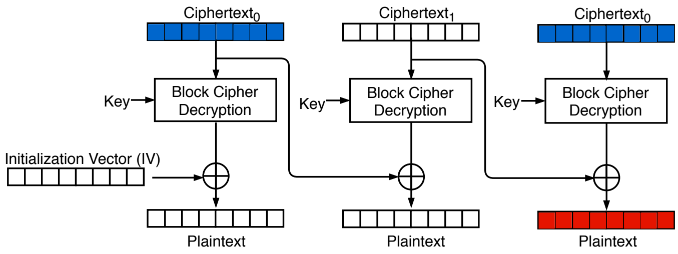

Figure 4: In the CBC mode, the decryption procedure is determined by both the current and previous ciphers. When the cipher at t_0 is replayed at time t_2 (blue), the decrypted plaintext can result in a fake but legal glucose reading.

defibrillators.

3.2.3. Bit-flipping Attacks

Another type of attack taking advantage of the CBC mode is bit(byte)-flipping. 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.

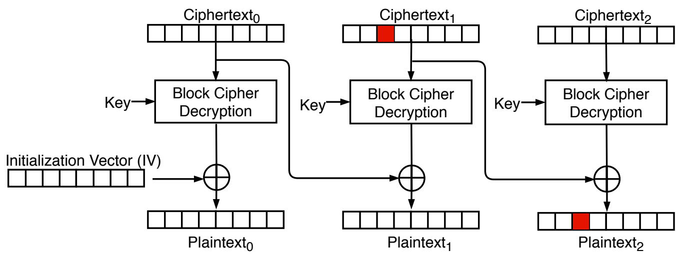

Figure 5: If a previous cipher is flipped by XOR operations in some bits, the next plaintext will be flipped accordingly by XOR operations in the same bits.

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

Example 3.2. In this example we show that by bit-flipping attacks at a 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 his/her meal and the glucose reading is at a high level, the patient will be ready to issue the insulin injection command, $\{0x00, 0x80\}$. If the attacker can predict this event, he can simply inject a forged command cipher at moment t_1 as:

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

It is notable that the attacker has a flexible choice of t_1, as long as it is 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 of insulin injection, after decryption in the CBC mode at moment t_2 the command becomes:

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

And it ends up with the command to *turn off* the device instead of insulin injection.

As for closed-loop devices such as defibrillators, this type of attack is not straightforward but is still possible. Since those IMDs only accept professional configuration from a clinic or a hospital, it demands that the attackers to have access to those professional devices or locations. However, these professional configuration devices can be found in many third party medical device trading websites such as MedWOW Ltd. [MedWOW (2018)]. This bit-flipping attack becomes viable which effectively allows them to turn off a defibrillator and put a patient in grave danger.

4. System Design Criteria

The goals of the proposed security system are to protect the IMD wireless channels from all the MITM attacks mentioned above while keeping a low power consumption overhead. In addition, the design should modify the current secure scheme on IMDs as little as possible, so that it can be smoothly adopted by the current IMD manufacturers.

4.1. Eavesdropping Resistant Properties

The current AES scheme is sufficient 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 **Definition 3.1 & 3.2**, it is possible for an attacker to eavesdrop and store a number of legal ciphers. Those two definitions suggest that an attacker may also be able to make a proper guess of the encrypted health data or commands (plaintext), since some of them are usually 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 should be involved.

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 blocks) 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 light weight 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 to the transmitted packets. Thus 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.

Secondly, each authenticated cipher should be valid only once in a lifetime. 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 mis-detection 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.

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 an overkill in terms of both security and power consumption and thus much less deployed. Each health 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 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.

The first few subsections provide the preliminaries of our proposed design. The last subsection is on the system introduction.

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 the 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 is to verify if the following equation holds:

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

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 verified that the right-hand side of the equation is always a non-zero polynomial of i or y of degree 1. It is easy to prove that for a certain message and an error e, the error missing probability 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 strong attacks. In strong attacks we assume that attackers have proper knowledge of the information part, the encoding scheme, and are able to issue any modifications to the message in the channel. It often cooperates 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.

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 ($h \le 2^b - 2$) 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

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

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:

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

The AMD decoder verifies if the following equation holds:

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

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:

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

It has been verified that the right-hand side of the equation is always a non-zero polynomial of x of degree 1. It is easy to prove that for a certain message and an error e, the error missing probability is at most:

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

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

Usually there is little error correcting code's (ECC) redundancy added to information part of the IMD sensors or controllers to restore the message from random errors. Since the proposed scheme should be encoded into 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 $\tilde{k} \in GF(2^{16})$ or less 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 authenticated encryption with MAC-then-Encrypt work flow. 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 can never 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 always). 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. At the sender side, the system first encodes the health data from the sensors or commands from the controller k into the 32-bit information part y, with a double-error correcting (ECC) OLSC code's redundancy r. The ECC is to ensure the reliability of the health data against random errors;

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. For Robust codes, x is always 1. This approach is to protect the authenticity of the information k;

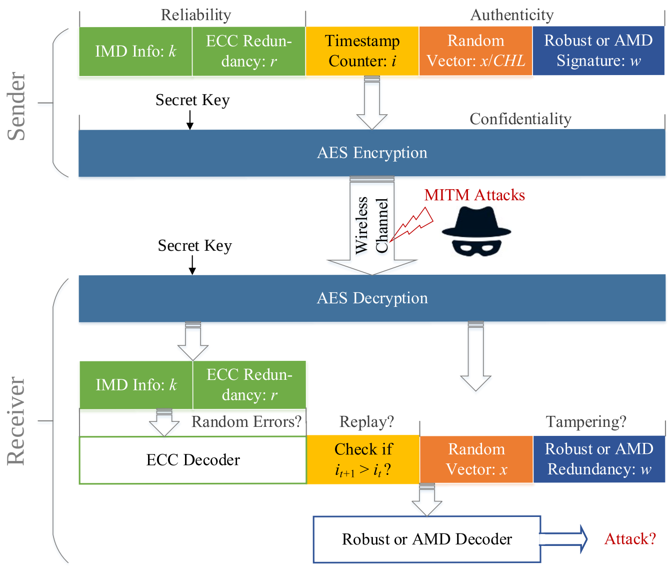

Figure 6: The entire system works under data blocks of size 128-bit with sub-block size 32-bit, or 192-bit block size with each sub-block in 48 bits. Such a block encapsulates authentication, obfuscation, timestamp, and random error correction to provide both reliability and security to the transmission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

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. In the wireless transmission channel, the Man-In-The-Middle attacks can occur by means of eavesdropping, hijack, replay, and bit-flipping etc.

5. On the receiver end, the AES module will decrypt the cipher;

6. The first IMD information and its redundancy are sent to the OLSC decoder for random error correction;

7. i is compared with the previous timestamp to check if it is up to date or a replayed message;

8. 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 attacks 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. Even under exactly the same circuit layout and manufacturing procedure, two IMD pieces will still have distinct behaviors under the same challenge.

The most commonly manufactured PUFs are memory PUFs and delay PUFs. The memory PUFs take advantage of the unpredictable boot-up value in each memory cell of a device. For example in SRAMs, before the cells are initialized, some of them always tend to be 1 and some to be 0 due to slight threshold voltage difference. For a given SRAM device, its memory boot-up readings are relatively stable each time, but different from all other devices due to their manufacturing variation. In this case the reading of the cells is the challenge, and the values of the cells are the response.

The delay PUFs amplify the intrinsic delay variance of among the gates or LUTs in a device, whose behavior also differs from device to device. In this case the challenge is usually the selection of the LUTs or gates to be compared, and the response is the comparison results (which element is faster).

Such a mechanism forms the unique "Silicon Fingerprints" for each device. PUFs are mostly used to verify the authenticity of a hardware device. However, in this paper we have it work for a novel purpose, which is 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) During a communication, say, an insulin injection command, when the controller/configuration device chooses to use x_j as the new random variable for the AMD encoding, it needs to update the IMD's AMD decoder about the choice. To do so, the controller/configuration device uses CHL_j instead of x_j in the encoded data block of Fig. 6;

(iii) The IMD applies CHL_j as its PUF input and generates locally the response x_j as the new random variable for decoding, and executes the command.

The above procedure is secure since CHL_j leaks zero knowledge of x_j. An attacker can never learn x_j in order to forge their 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 often the case in most existing attacks (error injection, forged command codes, replay etc.) - then the PUF with AMD codes is sufficient to secure the IMDs. CHL_j provides a private and secure channel for updating the variable x and learning CHL_j will not lead to acquiring x. Therefore, even if the AES module is not enabled, the forging a legal AMD signature is impossible.

Another advantage of using PUFs to facilitate the updating of x is that, each IMD is therefore uniquely linked 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 storage.

6. Third Party Authorization

There is always a need for the 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 of an IMD may have lost consciousness and be unable 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 is usually 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.

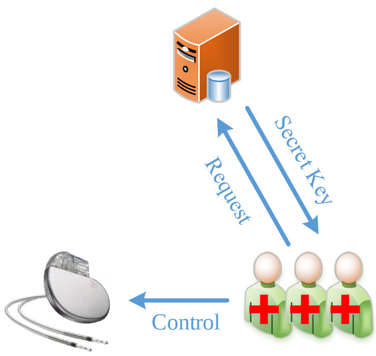

Figure 7: The access can involve the secret key or even the CRPs as mentioned in Remark 5.2.

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, he/she 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 staffers. 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 on 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, that only when the request level reaches t, can the secret key or other access credentials be authorized;

2. Each medical worker is given a request token with a request level based on their occupations;

3. When a medical worker request access, if the sum of their request level reaches t, the access to the IMD is granted. Below the t the IMD remains completely unaccessible.

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 i^{th} medical worker;
- h_i: the request token held by the i^{th} 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 [Pang and Wang (2005), Bu et al. (2017)]. All the computations should be carried out over Galois finite field (GF) arithmetic in order to maintain the information theoretic security. 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:

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

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:

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

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 enable different medical workers different levels of request privileges, based on their occupation or trustworthiness. Fig. 8 below illustrates a possible application scenario.

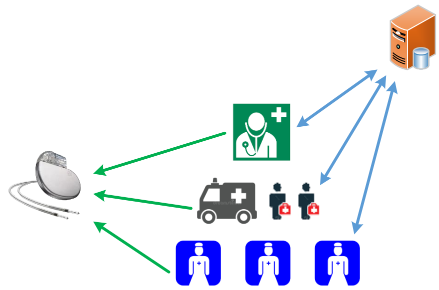

Figure 8: For example, a hospital can set the request threshold to $t = 3$ and allow surgeons or physicians to be entrusted with 3 request tokens while each EMT or paramedic on an ambulance is entrusted with 2 tokens and a nurse with 1. In case of an emergency, it will need at least 3 nurses to request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

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

At an emergency case the staff members with IDs $\{0x0003, 0x0005, 0x0007\}$ were sent out to a patient. Since $t = 3$, by substituting their IDs and shares into [Eq. 11], they are able to re-construct $A = 0x3F01$ and legally configure the patient's IMD.

If an attacker is able to bribe at most two staff members, say, $\{0x0004, 0x0006\}$, he/she still cannot break into the patient's IMD. This is because for 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 verify this probability we have run through tests on over 3 billions of simulated IMD radio transmissions on sensor's health data and controller's commands. This is about the total number of an ordinary IMD's transmissions in 10 years under a frequency of 10Hz. During the simulation the system mimics an IMD sending and receiving messages, and the attacker applying hijack, replay, and bit-flipping attacks alternatively in every around, while the receiver verifying 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 a strong security which not a single attack was missed among over three billions of attacks. Thus we also apply 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 result not only shows that the proposed protection scheme works well according to the theoretical estimation of 2^{-b} error mis-detection probability precisely, but also proves that the 32-bit and 48-bit schemes are secure enough for missing 0 attack under 3 billion of malicious hijack, replay, and bit-flipping attacks, providing sufficient security during the IMD's lifespan.

7.2. Power Consumption Overhead

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

The following 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 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 no more 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 an overkill to 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, comparing with 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 saves significant amount of the 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 IMDs' wireless communication. We prove theoretically and through experimental results that by authenticated encryption with a random vector and a timestamp encoded by Robust or AMD codes, it mis-detected 0 errors in a device's lifespan. Depending on the attack model, different authentication approaches can be used to achieve cost-efficiency. Robust codes with less hardware complexity are used when the attackers have little knowledge about the transmitted packets, while AMD codes with more cost to deal with the attackers having more knowledge on the packets. Moreover, the proposed authentication module's energy consumption is merely $3 \sim 4\%$ of the pre-installed AES module's. Compared with other authentication techniques such as HMAC, our approach consumes only 13% energy while providing the same security level. These advantages make the proposed scheme a secure and reliable solution to IMDs 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 know or be able to provide any information of the IMD device (secret keys or CRPs). Thus the third party medical team has to acquire it from the service point. To avoid intrusion, we design a threshold 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.

The power analysis of this paper is based on the AES module that has already been integrated in IMDs. However, since the Robust or AMD message authentication code is lightweight itself, it is recommended to also use a lightweight AES (or other encryption schemes) implementation to save battery life in IMDs.

9. Acknowledgments

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

References

Bu, L. and Karpovsky, M. (2016). 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*.

Bu, L. and Karpovsky, M. G. (2017). A design of secure and reliablewireless transmission channel for implantable medical devices. pages 233–242.

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

Missing \ b	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

I 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

I 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

I 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.

Bu, L., Nguyen, H. D., and Kinsy, M. A. (2017). 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*, pages 1–6.

Burleson, W., Clark, S. S., Ransford, B., and Fu, K. (2014). Design challenges for secure implantable medical devices. *Springer New York*.

Chede, S. and Kula, K. (2008). Design overview of processor based implantable pacemaker. *Journal of Computers*.

Cramer, R., Fehr, S., and Padró, C. (2013). Algebraic manipulation detection codes. *Science China Mathematics*,

56(7):1349–1358.
Daemen, J. and Rijmen, V. (2013). The design of rijndael: Aes-the advanced encryption standard. *Springer Science and Business Media*.
Gassend, B. et al. (2002). Silicon physical random functions. *Proceedings of the Computer and Communications Security Conference*.
Halperin, D., Heydt-Benjamin, T. S., Ransford, B., Clark, S. S., Defend, B., Morgan, W., Fu, K., Kohno, T., and Maisel, W. H. (2008). Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. *Proceedings of the 29th IEEE Symposium on Security and Privacy*.
InfoSec (2014). Hacking implantable medical devices. http://resources.infosecinstitute.com/hcking-implantable-medical-devices/.
Krawczyk, H. (2001). The order of encryption and authentication for protecting communications (or: How secure is ssl?). *Springer Berlin Heidelberg*.
Li, C., Raghunathan, A., and Jha., N. K. (2011a). 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*.
Li, C., Raghunathan, A., and Jha, N. K. (2011b). Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. pages 150–156. IEEE.
McDonald, J., Dean, S., Niewolny, D., Garcia, D., Chhabra, N., and Chang, L. (2011). *Integrated Circuits for Implantable Medical Devices*.
MedWOW (2018). http://www.medwow.com.
Neumeier, Y. and Keren, O. (2012). Punctured karpovsky-taubin binary robust error detecting codes for cryptographic devices. In *On-Line Testing Symposium (IOLTS), 2012 IEEE 18th International*, pages 156–161. IEEE.
Pang, L.-J. and Wang, Y.-M. (2005). A new (t, n) multi-secret sharing scheme based on shamirs secret sharing. *Applied Mathematics and Computation*, 167(2):840–848.
Paul, N., Kohno, T., and Klonoff., D. C. (2011). A review of the security of insulin pump infusion systems. *Journal of Diabetes Science and Technology*.
Radcliffe, J. (2011). Hacking medical devices for fun and insulin: Breaking the human scada system. *Black Hat Conference*.
Roberts, P. (2011). Blind attack on wireless insulin pumps could deliver lethal dose. http://threatpost.com/en_us/blogs/blind-attack-wirelessinsulin- pumps-could-deliver-lethal-dose-102711.
Rostami, M., Burleson, W., Koushanfar, F., and Juels, A. (2013). Balancing security and utility in medical devices? *Proceedings of the 50th Annual Design Automation Conference*.
Rushanan, M., Rubin, A. D., Kune, D. F., and Swanson, C. M. (2014). Sok: Security and privacy in implantable medical devices and body area networks. *2014 IEEE Symposium on Security and Privacy*.
Shamir, A. (1979). How to share a secret. *Communications of the ACM*.
Swepsie (2014). Bypassing encrypted session tokens using cbc bit flipping technique. http://swepssecurity.blogspot.com/2014/05/bypassing-encrypted-session-tokens.html.
Takahashi, D. (2011). Excuse me while i turn off your insulin pump. http://venturebeat.com/2011/08/04/excuse-mewhile- i-turn-off-your-insulin-pump/.
Tomashevich, V., Neumeier, Y., Kumar, R., Keren, O., and Polian, I. (2014). 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*, pages 40–45. IEEE.
Wang, Z. and Karpovsky, M. G. (2011). Manipulation detection codes and their application for design of secure cryptographic devices. *Proc of International Symposium on On-Line Testing (IOLTS)*.
Yalcin, G., Islek, E., Tozlu, O., Reviriego, P., Cristal, A., Unsal, O. S., and Ergin, O. (2014). Exploiting a fast and simple ecc for scaling supply voltage in level-1 caches. *IEEE International On-Line Testing Symposium (IOLTS)*.
Yury, C. (2014). Your heartbeat may soon be your only password. 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.

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.