Designing Novel Proxy-based Access Control Scheme for Implantable Medical Devices

Longfei Wu[a,*], John Du[b]

[a] *Department of Mathematics and Computer Science, Fayetteville State University, Fayetteville, 28301, NC, USA*
[b] *Methacton High School, 19403, Eagleville, PA, USA*

Abstract

Implantable medical devices (IMDs) are small medical devices implanted within the human body, performing diagnostic, monitoring, and therapeutic functions. Modern IMDs are equipped with a radio transmitter and can communicate with a specialized external programmer device (i.e., IMD programmer) through the wireless channel. IMDs are extremely limited in computation power, storage and battery capacity, hence can only afford lightweight cryptographic operations. This makes IMDs vulnerable to adversarial attacks especially on the wireless interface. In this paper, we propose a novel proxy-based fine-grained access control scheme for IMDs, which can prolong the IMD's lifetime by delegating the heavy cryptographic computations to a proxy device (e.g., smartphone). Additionally, we use the ciphertext-policy attribute-based encryption (CP-ABE) to enforce fine-grained access control so that only the qualified and/or authorized individuals can access the IMDs. The proposed scheme is implemented on real emulator devices. The experimental results show that the proposed scheme is lightweight and effective.

Keywords:
Implantable medical device (IMD), Access control, Proxy, Attribute-based encryption.

1. Introduction

The rapid global aging and unhealthy lifestyle in today's era have elevated the prevalence of chronic diseases not only among the elderly population but also the younger generation, resulting in the rise of burden on healthcare systems. Thanks to the advances in scientific and engineering technology, more and more patients can have long-term implantable medical devices (IMD) installed in their bodies to diagnose, monitor, and treat their conditions, diseases

*Corresponding author
Email address: lwu@uncfsu.edu (Longfei Wu)

and injuries, which can reduce the clinic and hospitalization needs and alleviate pressure on healthcare providers. For example, an insulin pump can monitor the glucose level and deliver insulin to treat diabetes, a pacemaker can regulate the beating of the heart using electrical impulses, and a neurostimulator can generate and send electrical impulses to the spine to treat chronic pain and movement disorders. Some IMDs are built for organ function replacement. For instance, a total artificial heart is implanted to take over the complete function of a patient's failing heart. According to a recent market research report published by Allied Market Research [3], the global IMD market is projected to reach $179 billion by 2030. In the United States, about 10% of Americans will have a device implanted into their bodies during their lifetimes [9, 31].

IMDs are equipped with an ultra-low-power radio transceiver to communicate with an external IMD programmer. The IMD programmer is the device used to collect medical data from IMDs and issue operation/configuration commands to IMDs for administering drugs, changing dosages, updating operation parameters, etc. Typically, an authorized medical staff in physical proximity or the patient themself can operate the IMD programmer to access the IMD.

The wireless communication and networking capabilities of IMDs are major sources of security vulnerabilities. Due to the broadcast nature of wireless channels, the physical signals exchanged between the IMD and the IMD programmer can be captured by eavesdroppers. The security vulnerabilities of IMDs can be exploited by adversaries to steal sensitive medical data, reset configuration parameters, and issue unauthorized commands to an IMD, which could result in fatal consequences. Unlike regular electronic devices that are computationally capable of public-key cryptographic operations and have sufficient power supply, IMDs have very limited computational power, storage, and battery capacity since they are embedded inside human body and must be small in size. There is some ongoing research on wireless charging technologies for IMDs [20, 22, 18, 39], but the influence of radiation and thermal dissipation in biological organs and tissues still needs years of testing and clinical trials before wireless charging can be widely implemented in commercial IMDs. Therefore, only lightweight cryptographic operations (e.g., symmetric cryptography, hashing) are considered practical for IMDs at the current stage.

In this paper, we propose a novel proxy-based access control scheme for IMDs, which can greatly reduce the computational overhead and power consumption of IMDs. The proxy communicates with the IMD programmer via an audio cable. Unlike USB connection and short-range wireless connections like Bluetooth and NFC, communications through headphone jack do not require the patient to unlock the phone for approval of connection, which is a practical concern for emergency situations in which the patient is unconscious. The proposed scheme employs the ciphertext-policy attribute-based encryption (CP-ABE) in order to provide a fine-grained access control over the qualifications of the programmer operator. In addition, our scheme provides accountability for the treatments given through IMDs in case of a medical dispute. We implemented our scheme on real emulator devices: the IMD is emulated by TelosB mote, the proxy is emulated by smartphone, and the IMD programmer is emu-

lated by Rasberry Pi 3. The evaluation results show that the proposed scheme is lightweight and effective.

2. Motivation

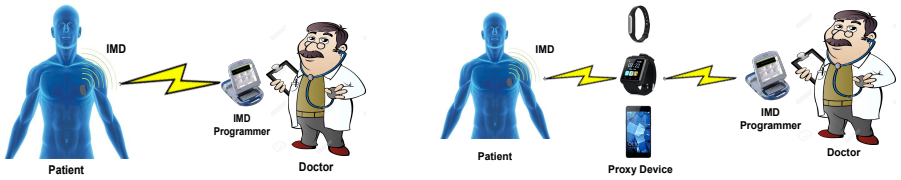
Existing research works have presented breaches in a number of commercial IMDs [15, 23, 27, 24, 4, 25], including the implantable cardioverter defibrillator (ICD), insulin pump and pacemaker. The U.S. Food and Drug Administration (FDA) provides resources and guidances [11] to help manufacturers design and maintain products that are cyber secure. The FDA also issues safety communications on the vulnerabilities found in commercial IMDs as well as the recommended actions patients, healthcare providers and manufacturers can take [10]. That is to say, although the FDA supervises and regulates the IMD industry, it only provides guidelines and recommendations for IMD security which are not legally binding. There is no standardized inspection on the security of commercial IMDs (software and hardware). The security and robustness of IMDs are reliant on the research and development teams of each individual manufacturers, who design access control schemes specific to their own products.

The security researchers have been seeking a generalized and effective access control scheme for all types of IMDs. In this section, we present the rationales for our proposed IMD access control scheme. Specifically, we first discuss some special but practical situations which can impact the functionality of IMD access control schemes, then we introduce the access control architecture and the access control model of our proposed IMD access control scheme.

2.1. Medical Emergency

IMD access control is not a difficult problem under regular situations, such as when a patient uses his/her own IMD programmer to access the IMD or when an acquainted physician uses hospital's IMD programmer to access the IMD in front of the patient. However, things become much more complicated in medical emergency situations. For example, when the patient falls sick while travelling out of town and needs immediate treatment, it is very likely that the patient faces an unknown IMD programmer operator who may be an emergency medical responder, a doctor, or a malicious attacker. Therefore, the patient has to verify the identity of the programmer operator as well as his/her qualifications to operate the patient's IMD. In a worse case, the patient may be unconscious due to illness and is not able to manually verify the programmer operator. To deal with such emergency cases, our proposed IMD access control scheme is able to:

1. verify the identity and qualifications of the programmer operator.
2. control the access autonomously without the patient's participation.



(a) Two-party Access Control (b) Proxy-based Access Control

Figure 1: IMD Access Control Architecture

2.2. Internet Connection

Online authentication has been widely used in IT applications. Intuitively, offloading authentication to the dedicated server of a governmental health agency or a hospital can greatly reduce the access control computations running on IMDs. This requires either the IMD (may be assisted by an external proxy) or the IMD programmer to be connected to the Internet. However, Internet access may not always be available, especially in depopulated areas with poor infrastructure. Hence, a robust IMD access control scheme should not rely on online authentication. Meanwhile, no Internet means that the access and the consequent treatments are not recorded in the online database, so the IMD programmer operator who has accessed the IMD is able to deny the access and the treatments administered in the case of medical dispute. Our proposed access control scheme can work without Internet connection and is able to hold the IMD programmer operator accountable for the provided treatments.

2.3. Access Control Architecture

The basic two-party access control architecture is composed of only the IMD and the IMD programmer, which is shown in Figure 1(a). Note that we view the IMD programmer and its operator together as one party since the operator needs to log in on the IMD programmer to be able to operate it. However, the basic two-party access control architecture is not suitable for resource-limited IMDs. Instead, we employ the proxy-based access control architecture which takes advantage of a proxy device to delegate the heavy cryptographic computations needed for access control. As shown in Figure 1(b), the proxy can be a smartphone or a wearable device (e.g., smart watch, smart bracelet) that is equipped with sufficient computational power and battery capacity.

The wireless communications between the IMD and the proxy are protected by the lightweight symmetric encryption, which is appropriate for IMDs in terms of both computational complexity and energy consumption. The IMD-to-proxy communications are considered secure given that a pair of symmetric keys have been distributed to the IMD and the proxy during their initial pairing. After that, the proxy device will perform the access control on behalf of the IMD.

Each time an IMD programmer attempts to access the IMD, the proxy needs to first build a communication channel with the IMD programmer. However, for general-purpose proxy devices like smartphones, setting up a local connection

Figure 2: Architecture of Access Control with Audio Cable Connection

with an external device (e.g. IMD programmer) via Bluetooth, NFC, or USB would require manual approval on the smartphone, which cannot be done when the patient is unconscious and the phone is locked by passwords/PINs. Therefore, we must choose a communication channel that is available even without the patient's involvement.

We found that most modern smartphones have a headphone jack/port (or a USB-C port/Lightning port that can be converted using an adapter) and most commercial IMD programmers have a USB port. Therefore, we propose that a smartphone serving as the proxy can be connected with an IMD programmer through an audio cable. As shown in Figure 2, the one end of the audio cable is plugged into the smartphone's headphone port while the other end is linked to an audio-to-USB adapter, which is then plugged into the USB port of the programmer. The mobile application for IMD access control is always running in the background, ready to process incoming IMD access requests from the audio cable.

The advantages of using the audio cable for communications include:

- **No patient involvement required.** The data can be transmitted through audio cable in a plug-and-play manner, even if the phone is locked.
- **Reduced attack surface.** The data exchanged for IMD access control are not exposed in the air. The nearby adversary cannot overhear the communications or actively interfere the access control procedure.
- **Low cost.** The audio cable connection does not require extra hardware. An audio-to-USB adapter only costs around $10.

The proxy-based access control depends on the presence of the proxy device. In the particular case that the proxy is not detected (lost or out of power), the commonly used strategy is that the IMD will enter the open-access mode in which it accepts all incoming access request [17, 38, 34, 8]. While this allows eligible physicians to still be able to access the IMD to provide treatments when the proxy is missing, it also gives adversaries opportunity to attack the IMD. The focus of our paper is to provide fine-grained access control with the proxy.

2.4. Access Control Model

In our scheme, the access control is performed by the proxy on behalf of the IMD, based on the information of the IMD programmer operator (i.e., identity,

qualifications) that has been transferred to the IMD programmer when the operator logs in on the programmer. The login can be made in either online or offline modes depending on the availability of Internet access and the preference of the programmer operator. In the offline mode, a smart card storing the operator's information is swiped on the programmer.

To achieve offline authentication of the programmer operator and hold the operator accountable for the treatment, the computation-intensive public-key cryptography must be used. The proxy device and the IMD programmer are powerful enough to run public-key cryptography. Specifically, the IMD programmer needs to provide the digital certificate of the programmer operator to the proxy, then the proxy validates the digital certificate. If the certificate is valid, the proxy will log the access request details (e.g., identity, time) signed by the programmer operator's private key as evidence of access.

In practice, not all physicians have the specialty and knowledge to operate a certain model of IMD programmer, hence an effective IMD access control scheme should be capable of checking the qualifications of the IMD programmer operator. The attribute-based access control (ABAC) model is desirable for this goal, in which the qualifications (medical specialty, eligibility to operate a certain model of IMD, etc.) have been assigned to the IMD programmer operator as attributes to enforce the access control. The access rule is defined as a mix of attributes, and the decision is made by matching the attributes required. The attribute-based encryption has been widely used in modern computing systems [35, 36, 37]. There are two major types of ABE schemes: Key-Policy Attribute-Based Encryption (KP-ABE) [13] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [5]. Both types of ABE schemes are collusion-resistant: colluding users cannot gain access by combining their associated attributes while none of them possesses the full set of required attributes by the access tree (i.e. access policy) [13][5]. In KP-ABE, the secret keys are generated from the access tree whose leaves are associated with attributes, and data are encrypted over a set of attributes. On the contrary, CP-ABE uses access tree to encrypt data and uses the set of attributes to generate the secret key. Both types of ABE schemes support the qualification checking of the programmer operator. Considering that the generation of secret key can be performed by a dedicated attribute authority, we should adopt the type of ABE scheme with faster encryption and decryption. Recent studies FAME [2] and FABEO [29] proposed fast-pairing based CP-ABE and KP-ABE, in which the decryption time is significantly reduced compared to encryption and does not depend on the number of attributes involved. Meanwhile, the encryption time for both CP-ABE and KP-ABE still scales near-linearly to the number of attributes (the amount of exponentiation operations is proportional to the number of attributes involved). Generally speaking, it is quite likely that a physician is qualified to operate more than just one IMD model and can even provide medical care in multiple fields (e.g., emergency medicine, family medicine, etc.), which means that the attributes required by an access policy tend to be a small subset of all attributes a qualified operator has. Therefore, we choose the CP-ABE scheme (encryption using the access policy) over the KP-ABE scheme (encryption over the full at-

tribute set). Specifically, the proxy encrypts a temporary symmetric session key with the access policy, and an IMD programmer can decrypt the ciphertext and obtain the session key to access the IMD if and only if its operator's attributes satisfy the access policy embedded in the ciphertext.

Additionally, to tackle the circumstance where the patient themself want to access their IMDs, each patient should be assigned a unique owner attribute which can satisfy the respective access policy enforced on their IMDs without needing the medical qualifications. The patient's owner attribute is stored in the patient's smart card. Like physicians, a patient also needs to log in on a programmer in either the online mode or the offline mode in order to access the IMD through the proxy. The patients would be accountable for their own operations performed to their IMDs.

3. Protocol Design

3.1. System Overview

- **IMD.** Each IMD has a unique identification ID_i and a master key K_i^M. Note that the master key is only used for pairing up the IMD with a proxy device, and will not directly participate the access control procedure.

- **Proxy.** The proxy device has the identification ID_p. There is a client program running on the proxy to perform the access control for the IMD. The proxy has been paired up with the IMD through initial setup. The client program has a copy of the public parameters PK used to run CP-ABE, and is able to generate ciphertext with the access tree (policy) T which describes the qualifications required for access.

- **Operator.** All legitimate programmer operators must first be registered at a Central Health Authority (CHA), which manages the qualifications of operators and issues digital certificates for them. After registered at CHA, each programmer operator is assigned a unique identification ID_o, a pair of public/private keys KU and KR, and a digital certificate $Cert$. Additionally, the qualifications that an operator has correspond to a set of associated attributes S, and the secret key SK for CP-ABE decryption is generated based on the set of attributes S.

- **Programmer.** The IMD programmer can be simply viewed as the terminal device used by its operator to interact with the IMD or proxy. It obtains all information required for access control (e.g., digital certificate, parameters for CP-ABE) from the operator, by online login (ahead of IMD access control when Internet is available) or reading from the operator's token (e.g., smart card) in the offline mode. Once an access session terminates, the operator's information will be erased by the programmer.

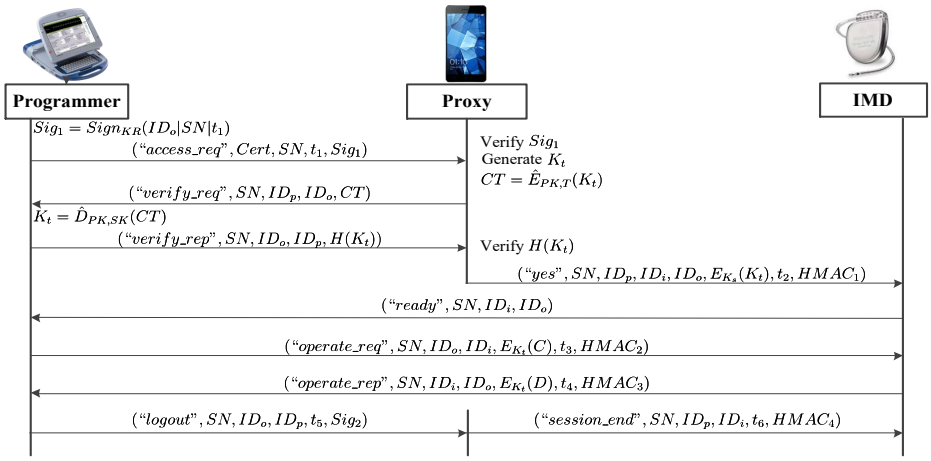


Figure 3: Access Control Procedure

3.2. Access Control Protocol Design

The access control procedure is presented in Figure 3. We assume that the proxy device has already been paired up with the IMD, and a pair of symmetric keys K_s have been shared between them for encrypted communications. The detailed procedure is described as follows:

1. The programmer initializes the access control protocol by connecting to the proxy via audio cable, and sending an access request which is composed of a unique action sequence "*access_req*", the operator's digital certificate $Cert$, a random selected session number SN, timestamp t_1, and a signature Sig_1 signed by the operator's private key KR. The certificate contains the operator's public key KP and identification ID_o. The signature $Sig_1 = Sign_{KR}(ID_o|SN|t_1)$ is attached to prove that the current access requestor is indeed ID_o.
2. The access control mobile application has registered a receiver of the headset connection state changes. When this program is notified of the plug-in event, it will read in and demodulate the audio data. If the action sequence "*access_req*" is detected in the demodulated data, it will indicate that the data is for IMD access request instead of regular audio (e.g., music). Then it extracts and verifies the received Sig_1 using the requestor's certified public key KP embedded in $Cert$. If the signature is valid, the programmer operator is successfully authenticated. The proxy will next check if that operator is authorized to access. Specifically, it randomly generates a temporary session key K_t and encrypts K_t with the required access policy T using CP-ABE. Then, the produced ciphertext $CT = \hat{E}_{PK,T}(K_t)$ is sent back to the programmer.
3. The programmer decrypts CT with the operator's secret key SK. If the operator's qualifications (attributes) satisfy the required access policy T,

the temporary session key K_t can be obtained. Then, it calculates the hash value of K_t and sends the hash value to the proxy.

4. The proxy also calculates the hash value of K_t with the same hash function. If the two hash values are equal, it indicates that the programmer operator is eligible for access. The proxy will inform the IMD that the programmer ID_o has been authenticated and is authorized to access, and sends a copy of the session key K_t to IMD encrypted by. Note that all communications to/from the IMD are conducted in the wireless channel, which may suffer eavesdropping, replay, and tampering attacks. Hence, the session key K_t is encrypted by the shared key K_s to prevent eavesdropping. A timestamp t_2 is added to defend replay attacks. The keyed-hash message authentication code (HMAC) of the message is attached using K_s to ensure the authenticity and integrity of the message.
5. After receiving the authorization notification from the proxy, the IMD retrieves the session K_t and sends "ready" message to the programmer.
6. In the mutual communications between the IMD and the programmer, the operation commands C sent by the programmer and the data/result D returned by the IMD are all encrypted using the temporary session key K_t. Each authorization permits multiple operations (e.g., data reading, drug delivery). We only draw one round of operation in Figure 3 for illustration. Similarly, the timestamps and HMACs are adopted in the communications between the IMD and the programmer to defeat various active wireless attacks.
7. After the programmer has completed the access, it sends a "logout" notification message to the proxy. Another signature $Sig_2 = Sign_{KR}(ID_o|SN|t_5)$ is generated in which the access end time t_5 is signed.
8. Finally, the proxy will notify the IMD that the current session has ended so that the session key K_t will also expire. Timestamp t_6 and $HMAC_4$ are included in this message.

Our scheme asks the programmer to explicitly log out of the session, and requires it to sign the time that session ends. Therefore, Sig_1 and Sig_2 together can verify that the operator has accessed the IMD in that period of time. Any wrong treatment performed in this period can be attributed to that operator. However, it may happen that the programmer does not sign out by the end of its session. A time-out mechanism is used to address this situation. Specifically, if there is no interaction made by the programmer for a fixed amount of time T_{out}, the session will be closed and the session key K_t will be disabled. Additionally, the programmer will delete all the operator's data at the end of the session.

4. Security Analysis

4.1. Resistance to Passive Attacks

In the wireless channel, the adversary can eavesdrop the communications between the proxy and the IMD, as well as between the programmer and the

IMD, in order to obtain the operation commands, data/results, and session keys being transmitted. To defend against these passive wireless attacks, all sensitive information is protected with symmetric encryptions. Specifically, a pair of keys K_s are pre-shared by IMD and the proxy during initial pairing, and another pair of keys K_t are securely distributed to the IMD and the programmer. The adversary cannot decrypt the ciphertexts without knowing these keys.

4.2. Resistance to Active Attacks

- **Masquerade attack.** The attackers may attempt to impersonate an authorized entity to access the IMD. Specifically, they may purchase or steal an IMD programmer device, but they cannot access the IMD since they do not own the attributes required by the access policy. We assume that they cannot steal the smart cards of a physician or patient.

- **Replay attack.** When the IMD programmer and the IMD communicate, all messages containing the treatments to be administered and the returned data/result are timestamped to prevent replay attacks.

- **Tampering attack.** The messages transmitted over the wireless channel are vulnerable to malicious modifications. To prevent attackers from tampering the sensitive information in the messages like keys, treatment commands, returned data/results, HMACs are calculated with the shared keys used for symmetric encryption (K_s for messages between the proxy and the IMD and K_t for messages between the programmer and the IMD). Since attackers do not have these keys, they will not be able to generate valid HMAC for the tampered message and the tampering will be exposed.

- **Denial-of-service Attack.** The attackers may launch DoS or DDoS attacks to prevent an authorized entity to access the IMD. At the remote end, they could disrupt the web server so that operators cannot log in on the IMD programmer in the online mode. Our scheme allows the operators to swipe their smart cards to log in instead. The communications between the IMD programmer and the proxy are conducted locally through the audio cable, hence are resistant to DoS/DDoS attacks. However, since IMD can only communicate wirelessly with the proxy or a programmer device, the attackers can launch DoS attack towards an IMD by jamming its wireless communication. The common solution includes spectrum spreading, channel hopping, etc. This issue is beyond the scope of this paper.

5. Evaluation

In this section, we implement our schemes with real devices and evaluate the overheads of our scheme with experiments.

Figure 4: Prototype Setup

5.1. Prototype Implementation

The major challenge in the implementation of our scheme is the difficulty to find open-source commercial IMD products. Alternatively, in our prototype system, we use TelosB Mote TPR2420 sensor node with 8 MHz TI MSP430 microcontroller, 10kB RAM, and 48kB Flash Memory to emulate the IMD. We choose the Rasberry Pi Version 3 Model B, a small single-board device with 1.2GHz 64-bit quad-core CPU and 1GB RAM, to emulate the programmer. A Nexus 4 smartphone powered by a 1.5 GHz quad-core processor with 2 GB of RAM is used as the proxy.

In our scheme, symmetric encryption and public-key cryptography are implemented using the Advanced Encryption Standard (AES) algorithm and RSA algorithm, respectively. We use 128 bits key for AES encryption and 1024 bits keys for RSA encryption. SHA-1 is chosen for HMAC generation.

5.2. Testbed Specification

We developed an Android application for Nexus 4 smartphone, which can perform CP-ABE encryption and modulate/demodulate audio signals. Also, we developed a client program running on Raspberry Pi, which can perform CP-ABE decryption and process digital audio data. The smartphone and the programmer are connected via an audio cable and a USB Sound Adapter (audio-to-USB convertor). A screenshot of the prototype is shown in Figure 4.

We adopted the Binary Frequency-Shift Keying (BFSK) frequency modulation scheme for modulation, in which the digital data are converted into the analog signals at two different frequencies for transmissions over the audio cable. For example, the binary "0" bit is represented by the audio signal at frequency f_0 while the binary "1" bit is represented by the audio signal at frequency f_1. At the receiver end, the analog signal is sampled to a sequence of discrete-time signal (samples). Then, we use the Discrete Fourier Transform (DFT) algorithm to convert the sampled analog signal from time domain into the frequency domain representation. Specifically, the analog signal to be demodulated can be viewed as an addition of multiple sine signals in different frequencies. With

Table 1: Parameters for modulation and demodulation

Parameter	Value
Sampling rate	44100Hz
Pulse-code modulation (PCM) bit depth	16
f_0	1575Hz
f_1	3150Hz
Baud rate	315

Fourier transform, the magnitudes of the modulated signal on various frequencies within the spectrum range are calculated. The frequency with the highest amplitude (i.e., maximum power) is called the peak frequency. If the peak frequency equals f_0, then the current signal sample represents a "0" bit; while if the peak frequency equals f_0, then the signal sample represents a "1" bit. The parameters we used for modulation and demodulation are listed in Table 1.

5.3. Experimental Results

To evaluate the efficiency of our scheme, we measure the computational overheads of the protocols running on the IMD (TelosB Mote), the proxy (Nexus 4 smartphone), and the IMD programmer (Rasberry Pi), respectively. All the run-time overheads are the average of 50 measurements.

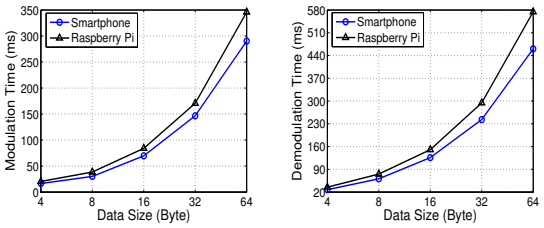
5.3.1. IMD

The major cryptographic computations performed on the IMD are the symmetric encryption and HMAC. On TelosB node, the 128-bit AES encryption takes 2ms. For HMAC, we estimate the length of the plaintext message ("$operate\_rep$", $SN, ID_i, ID_o, E_{K_t(D)}, t_4$) to be 78 bytes in total, including a 4-byte command, a 2-byte sequence number, two 4-byte IDs, and a 64-byte data/result returned). The HMAC computation over a 78-byte message takes 47ms.

5.3.2. IMD programmer and proxy

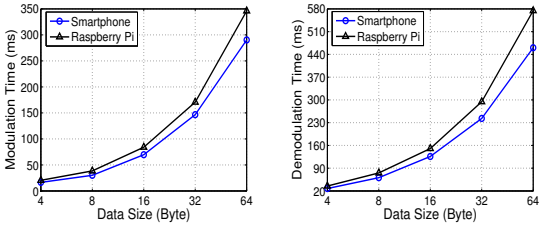
The IMD programmer (Rasberry Pi) and proxy (smartphone) both need to conduct modulation/demodulation for their wired communications through the audio cable. Figure 5(a) and Figure 5(b) show the time consumption for modulation and demodulation on smartphone and Rasberry Pi, respectively. As we can see, the Rasberry Pi has a better performance than the smartphone, and the demodulation takes longer than the modulation process.

Additionally, the proxy (smartphone) needs to encrypt the temporary session key with CP-ABE encryption algorithm, and the IMD programmer (Rasberry Pi) will run the decryption algorithm to retrieve the key. Figure 6(a) and Figure 6(b) show the time consumption for CP-ABE encryption on smartphone and decryption on Rasberry Pi, respectively. The run-time overheads for CP-ABE encryption and decryption both increase with the number of leaf nodes (attributes). Our experiments tested a maximum of 20 attributes, which should be sufficient to specify the qualifications of the programmer operator. The decryption is found to be the most time-consuming step in the whole scheme,



(a) Modulation (b) Demodulation

Figure 5: Time Consumptions - Modulation and Demodulation



(a) CP-ABE Encryption (b) CP-ABE Decryption

Figure 6: Time Consumptions - CP-ABE Encryption and Decryption

which takes about twice the time used for encryption. Since the CP-ABE encryption/decryption is required only once, their execution time are considered acceptable.

6. Related Work

Existing IMD access control schemes are limited due to the practicability or the granularity of access control.

Most of the existing schemes employ the two-party access control architecture, in which the resource-constrained IMD performs the access control by itself, resulting in a relatively weak access control and shorter IMD lifetime. Some works proposed that the IMD should be accessible only to a group of trusted people (e.g., physicians, relatives) who must be added into the IMD's access control list (ACL) [33, 14] ahead of time. Some other works assume that the IMD and the programmer have a pre-shared secret like a master key [15] or rolling code [23]. However, these schemes are not practical considering that the IMD may be accessed by any physicians at the place that the patient travels to, who have not been added to the IMD's ACL or had any pre-shared secret. Another type of IMD access control relies on authentication server [26, 12, 32, 1], which is constrained by the availability of Internet access. The most popular category of existing schemes grants access based on whether the programmer is in proximity of the IMD. Specifically, the IMD and the programmer need to

extract features from the same source simultaneously. If they are close enough, they can generate the same temporary key from the extracted features and use that key for the encryption of future communications. The source is usually a signal in an out-of-band channel, such as electrocardiography (ECG) signal [30, 6], body-coupled electric signal [7], vibration [21], ultrasound [28], NFC [19, 16], etc. However, the common disadvantage of these proximity-based access control schemes is that they only provide a coarse-grained access control - they cannot recognize who is accessing the IMD and an attacker with a purchased or stolen IMD programmer can also gain access.

By contrast, in the proxy-based IMD access control schemes, the powerful proxy device can support more complicated and fine-grained access control schemes. However, existing proxy-based access control schemes either depend on the anomaly detection of access pattern [17, 38] or only verify the authenticity of the IMD programmer instead of its operator [34, 8]. Both types of schemes do not check the IMD programmer operator. This could be dangerous since an attacker can also purchase or steal a legitimate IMD programmer. Our scheme views the IMD programmer and its operator together as one party, and the access decision is made based on the programmer operator's information stored in the programmer. We not only authenticate the operator, but also employ the CP-ABE algorithm to verify the qualifications of the operator, hence providing a fine-grained access control. Moreover, our proposed scheme can ensure the accountability for the actions made by the operator.

7. Conclusion

In this paper, we proposed a novel fine-grained IMD access control scheme based on a proxy device like the patient's smartphone, which will delegate the heavy access control computations for the IMDs. The communications between the proxy and the IMD programmer are conducted through an audio cable, which does not need the patient to manually approve their connection. The ciphertext-policy attribute-based encryption is employed to enforce the fine-grained access control over the qualifications of the programmer operator. We built a prototype to evaluate our scheme. The experimental results demonstrated its feasibility and effectiveness.

8. ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) under grant No. 1901010.

References

[1] Aghili, S.F., Mala, H., Shojafar, M., Peris-Lopez, P., 2019. Laco: Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in iot. future generation computer systems 96, 410–424.

[2] Agrawal, S., Chase, M., 2017. Fame: Fast attribute-based message encryption, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 665–682.

[3] Allied Market Research, 2017. Implantable medical devices market by product: Global opportunity analysis and forecast, 2014 - 2022.

[4] B. Jack, 2013. Implantable medical devices: Hacking humans. https://en.wikipedia.org/wiki/Barnaby_Jack\#cite_note-medcity-11.

[5] Bethencourt, J., Sahai, A., Waters, B., 2007. Ciphertext-policy attribute-based encryption, in: Proc. of IEEE S&P.

[6] Camara, C., Peris-Lopez, P., De Fuentes, J.M., Marchal, S., 2020. Access control for implantable medical devices. IEEE Transactions on Emerging Topics in Computing 9, 1126–1138.

[7] Chang, S.Y., Hu, Y.C., Anderson, H., Fu, T., Huang, E.Y.L., 2012. Body area network security: Robust key establishment using human body channel, in: Proc. of USENIX HealthSec.

[8] Denning, T., Fu, K., Kohno, T., 2008. Absence makes the heart grow fonder: New directions for implantable medical device security, in: Proc. of USENIX HotSec.

[9] EDITOR, 2021. Implantable material and device regulation. AMA Journal of Ethics 23. URL: https://journalofethics.ama-assn.org/issue/implantable-material-and-device-regulation.

[10] FDA , . Cybersecurity safety communications and other alerts. https://www.fda.gov/medical-devices/digital-health-center-excellence/cybersecurity#safety.

[11] FDA, . Cybersecurity guidances. https://www.fda.gov/medical-devices/digital-health-center-excellence/cybersecurity#guidance.

[12] Fu, C., Du, X., Wu, L., Zeng, Q., Mohamed, A., Guizani, M., 2019. Poks based secure and energy-efficient access control for implantable medical devices, in: International Conference on Security and Privacy in Communication Systems, Springer. pp. 105–125.

[13] Goyal, V., Pandey, O., Sahai, A., Waters, B., 2006. Attribute-based encryption for fine-grained access control of encrypted data, in: Proc. of ACM CCS.

[14] Halperin, D., Heydt, T., Fu, K., Kohno, T., Maisel, W., 2008a. Security and privacy for implantable medical devices. IEEE Pervasive Computing .

[15] Halperin, D., Heydt, T., Ransford, B., Clark, S., Defend, B., Morgan, W., Fu, K., Kohno, T., Maisel, W., 2008b. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses, in: Proc. of IEEE S&P.

[16] Hei, X., Du, X., Lin, S., 2014. Poster: Near field communication based access control for wireless medical devices, in: Proc. of ACM MobiHoc.

[17] Hei, X., Du, X., Wu, J., Hu, F., 2010. Defending resource depletion attacks on implantable medical devices, in: Proc. of IEEE GLOBECOM.

[18] Khan, S.R., Pavuluri, S.K., Cummins, G., Desmulliez, M.P.Y., 2020. Wireless power transfer techniques for implantable medical devices: A review. Sensors 20.

[19] Kim, B., Yu, J., Kim, H., 2012. In-vivo nfc: Remote monitoring of implanted medical devices with improved privacy, in: Proc. of ACM SenSys.

[20] Kim, D., Jeong, D., Kim, J., Kim, H., Kim, J., Park, S.M., Ahn, S., 2020. Design and implementation of a wireless charging-based cardiac monitoring system focused on temperature reduction and robust power transfer efficiency. Energies 13.

[21] Kim, Y., Lee, W.S., Raghunathan, V., Jha, N.K., Raghunathan, A., 2015. Vibration-based secure side channel for medical devices, in: Proc. of IEEE Design Automation Conference (DAC).

[22] Lee, H., Jung, S., Huh, Y., Lee, J., Bae, C., Kim, S.J., 2021. An implantable wireless charger system with 8.91 increased charging power using smartphone and relay coil, in: 2021 IEEE Wireless Power Transfer Conference (WPTC), pp. 1–4.

[23] Li, C., Raghunathan, A., Jha, N.K., 2011. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system, in: Proc. of IEEE HealthCom.

[24] Marin, E., Singelée, D., Yang, B., Verbauwhede, I., Preneel, B., 2016. On the feasibility of cryptography for a wireless insulin pump system, in: Proc. of ACM CODASPY.

[25] Marin, E., Singelée, D., Yang, B., Volski, V., Vandenbosch, G.A., Nuttin, B., Preneel, B., 2018. Securing wireless neurostimulators, in: Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, pp. 287–298.

[26] Park, C.S., 2014. Security mechanism based on hospital authentication server for secure application of implantable medical devices. Hindawi BioMed Research International .

[27] Radcliffe, J., 2011. Hacking medical devices for fun and insulin: Breaking the human scada system, in: Black Hat USA.

[28] Rasmussen, K., Castelluccia, C., Heydt, T., Capkun, S., 2009. Proximity-based access control for implantable medical devices, in: Proc. of ACM CCS.

[29] Riepel, D., Wee, H., 2022. Fabeo: Fast attribute-based encryption with optimal security, in: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 2491–2504.

[30] Rostami, M., Juels, A., Koushanfar, F., 2013. Heart-to-heart (h2h): authentication for implanted medical devices, in: Proc. of ACM CCS.

[31] Salazar, L., 2021. Addressing the medical device safety crisis. The Regulatory Review .

[32] Siddiqi, M.A., Doerr, C., Strydis, C., 2020. Imdfence: Architecting a secure protocol for implantable medical devices. IEEE Access 8, 147948–147964.

[33] Spring, R., Freudenthal, E., Estevez, L., 2007. Practical techniques for limiting disclosure of rf-equipped medical devices, in: IEEE Dallas Engineering in Medicine and Biology Workshop.

[34] Xu, F., Qin, Z., Tan, C.C., Wang, B., Li, Q., 2011. Imdguard: Securing implantable medical devices with the external wearable guardian, in: Proc. of IEEE INFOCOM.

[35] Yu, Y., Li, Y., Yang, B., Susilo, W., Yang, G., Bai, J., 2020a. Attribute-based cloud data integrity auditing for secure outsourced storage. IEEE Transactions on Emerging Topics in Computing 8, 377–390. doi:10.1109/TETC.2017.2759329.

[36] Yu, Y., Shi, J., Li, H., Li, Y., Du, X., Guizani, M., 2020b. Key-policy attribute-based encryption with keyword search in virtualized environments. IEEE Journal on Selected Areas in Communications 38, 1242–1251. doi:10.1109/JSAC.2020.2986620.

[37] Yu, Y., Xue, L., Li, Y., Du, X., Guizani, M., Yang, B., 2018. Assured data deletion with fine-grained access control for fog-based industrial applications. IEEE Transactions on Industrial Informatics 14, 4538–4547. doi:10.1109/TII.2018.2841047.

[38] Zhang, M., Raghunathan, A., Jha, N.K., 2013. Medmon: Securing medical devices through wireless monitoring and anomaly detection. IEEE Transactions on Biomedical Circuits and Systems 7.

[39] Zhou, Y., Liu, C., Huang, Y., 2020. Wireless power transfer for implanted medical application: A review. Energies 13.