THE BLE CLOAKER: SECURING IMPLANTABLE MEDICAL DEVICE COMMUNICATION OVER BLUETOOTH LOW ENERGY LINKS

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Taylor Anthony Nesheim

September 2015

COMMITTEE MEMBERSHIP

TITLE:	The BLE Cloaker: Securing Implantable Medical Device Communication over Bluetooth Low Energy Links
AUTHOR:	Taylor Anthony Nesheim
DATE SUBMITTED:	September 2015
COMMITTEE CHAIR:	Zachary N J Peterson, Ph.D. Assistant Professor of Computer Science
COMMITTEE MEMBER:	Phillip L. Nico, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	John Seng, Ph.D. Assistant Professor of Computer Science

ABSTRACT

The BLE Cloaker: Securing Implantable Medical Device Communication over Bluetooth Low Energy Links

Taylor Anthony Nesheim

Historically Implantable Medical Devices (IMDs) such as pacemakers have only been able to communicate to external devices through close proximity means of communication, primarily through inductive telemetry. Because of the unlikelihood of an adversary being able to gain access to an IMD through this type of communication, these devices were never designed with security in mind. However the recent advent of IMDs that are equipped with long-range wireless capabilities has made it necessary to consider how to secure these devices from malicious attacks.

This work presents an implementation of prior work that developed a theoretical security model whose specific intent was to secure IMDs with long-range wireless capabilities against both passive and active adversaries, while also ensuring the safety of the patient. This implementation is known as the Bluetooth Low Energy (BLE) Cloaker model and provides a prototype system that uses BLE as the long-range communication medium between an emulated IMD, an external programmer, and the BLE Cloaker device itself. The BLE Cloaker acts as a secure data proxy between the IMD and the external programmer. This prototype shows the benefits and drawbacks of this theoretical model when used in a real world system as well as the security strengths and weaknesses of using BLE as the wireless link in a medical application.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

CHAPTER 1 – Introduction

Until recently, Implantable Medical Devices (IMDs) were only able to communicate to external devices through close-proximity methods such as RF inductive telemetry. The necessity of being in such close-proximity made the possibility of a malicious adversary attacking these devices extremely unlikely. As such, the security of IMDs was never considered when they were originally designed. Modern IMDs however are equipped with long-range wireless communication that ranges from a few feet to across the room. Because of the swiftness of this change coupled with the previous lack of a need for it, IMDs are now completely unsecured against both passive and active adversaries. The purpose of this work is to provide a communications protocol for securing IMDs against cyberattacks. It additionally aims to provide and test an implementation of this communications protocol using off-the-shelf hardware.

1.1 Statement of Research Question

This work aims to answer the following research questions: what kind of communication protocol is needed for an Implantable Medical Device (IMD) equipped with Bluetooth Low Energy and RFID technology to be secured against malicious intrusion? Because of the current lack of security present in these IMDs combined with the life threatening consequences of device failure, ongoing research to find a solution to this problem is both important as well as urgent. There are several important criteria that need to be kept in mind when developing this kind of protocol. This protocol needs to provide protection from passive adversaries that have the ability to eavesdrop on wireless communications from the IMD, thereby gaining access to private patient data. It also needs to provide protection from active adversaries that have the ability to employ Denial

of Service attacks, Replay attacks, and device spoofing that could lead to improper functionality that could be harmful or fatal to the patient. It needs to ensure that the IMD is able to be accessed immediately by medical practitioners in a medical emergency, but is secured against malicious adversaries during all other times. Any protocol implemented needs to ensure proper device operation in a low power and low resource environment. This means that the battery life of the IMD must be conserved as much as possible as the replacement of depleted devices poses a significant risk to the well-being of the patient.

1.2 General Approach

The general approach that will be used to address this problem will be to design and develop a communications protocol that meets the above requirements based on a review of what the current best practices are for security in terms of encryption and authentication. Once the design has been completed, it will be implemented in an iterative process on actual hardware. The first iteration will consist of getting the various components setup and talking to each other with a focus on function rather than on security. The next iteration will involve refactoring the implementation to incorporate the aforementioned security including communication encryption and device authentication. The final iteration will refine the previous iterations as needed in order to achieve the constraints that will be present in an actual IMD. The implementation will then be put through various experiments to test and verify that it is conforming to the real world constraints while maintaining the necessary level of security.

CHAPTER 2 – Background and Related Work

2.1 Background

The notion of the development and use of Implantable Medical Devices is not a new concept. Doctors first considered the idea in the early 50s, and in 1952 the first external pacemaker was developed [4]. It was wall powered and caused a painful burning of the skin when in use, but it was a first step towards the devices that are commonplace today. With the advent of the transistor and the lithium battery, much smaller devices were possible and the further development of printed circuit boards and microchips in the 1980s led to the widespread development and implantation of pacemakers and other IMDs that closely resemble those that are seen today [4].

Historically, the primary method of communications of these devices with external programmers has been through RF inductive telemetry. Security was never an issue nor a primary concern of medical device companies because the only means for interfering with the device was through close proximity to the patient using the RF inductive telemetry mentioned above. Now however, more and more devices are being equipped with long-range wireless capabilities. This fact has substantially increased the possibility for adversaries with malicious intent to interfere with the proper functioning of these devices. When combining this with the knowledge that security has never been a primary concern of medical device companies, the potential for current and future device failures due to adversarial attacks is a huge concern. This also justifies the need for both the research in the area of IMD security and the subsequent implementation of this research by medical device companies in their products.

There have been several research papers written in recent years that specifically highlight the inherent security vulnerabilities in currently used models of IMDs. One such paper written by Halperin, Daniel, et al. shows that they were able to both read from and modify settings of an Implantable Cardioverter Defibrillator (ICD) using an off-the-shelf RF Commodity Software Radio [5]. The method they employed was simply to use this RF Radio to attempt to passively eavesdrop on known commands between the ICD and an external programmer, and then try to replay these communications back to the ICD and verify whether or not the ICD accepted or rejected the command. While doing this they found several worrisome results. The first was that both the eavesdropped patient and medical data being transmitted were completely in plaintext, making it straightforward to decode the information without even needing to know anything about the underlying structure of the communication packets. The second was that they were able to successfully replay previously eavesdropped packets and have the ICD accept these commands within several attempts. With just these two simple attacks, they were able to make both small changes such as modifying the patient name stored in the device, and much more devastating changes including disabling device therapies and inducing fibrillation that would send shocks of approximately 138V to a patient.

At first glance, the obvious solution is for medical device companies to simply adopt the industry best practices for security and apply them to all of their medical devices. Unfortunately, medical devices have significantly different requirements than most other software and hardware systems on the market. Because of this, industry best practices are either not applicable or need to be modified in order to be feasible in IMDs. In another paper written by Halperin, Daniel, et al., they examine the various system

requirements inherent to IMDs as well as go through the tradeoffs between these requirements and enforcing security [6]. These tradeoffs are summarized as follows:

- **Security vs. Accessibility:** IMDs need to be made secure against all unauthorized users. However these devices need to be made available to surgeons and healthcare professionals during critical and emergency situations. Failure to do so could cause harm or death to the patient or those attempting to operate on the patient. Adding a software back door is one possible solution to this, however this means intentionally adding an exploitable security vulnerability to the IMD.
- **Security vs. Device Resources:** The large majority of IMD devices are extremely resource limited due to their size and battery longevity requirements. Standard cryptographic methods of security are computationally expensive and are therefore not practical for use in resource constrained devices such as IMDs. Protecting against Denial of Service (DoS) attacks are of the utmost importance in these devices especially those relating to battery depletion and buffer overflow attacks. Any DoS attack could necessitate the explanting of the IMD from the patient, which increases the likelihood of infection and patient injury.
- **Security vs. Usability:** The user interfaces associated with Clinical Programmers and the devices themselves need to maintain security while also ensuring usability. If the usability of a programmer is such that a doctor or patient is unable to properly update or change settings on the IMD, it limits accessibility to that device and can again lead to adverse consequences for the patient. Likewise, although patient comfort and convenience is a priority, it can also come at the cost of security. Long range RF wireless capable IMDs for example allow patients a greater sense of comfort and

freedom as they are able to move about their home or doctor's office unconstrained. But this wireless capability comes at the cost of increasing the likelihood of passive and active attacks from adversaries.

Because of these and other issues uniquely attributed to the requirements of IMDs, securing these devices against malicious attacks is by no means a straightforward or trivial task. Each class of IMD has their own set of unique requirements and constraints that need to be considered separately when determining how to secure a particular device. There are no shortcuts or silver bullets in security, and when lives are at stake software developers need to do their due diligence to ensure that they have done all they can to protect patients against security vulnerabilities.

2.2 Related Work

This research is based upon previous work in which the authors outlined an IMD security model that incorporates a new type of device they deemed *The Cloaker* [3]. The model they put forth was mostly theoretical and involved only a simulation in software. Although this is a good first step, their work would benefit from an actual implementation under realistic hardware and software constraints found in typical IMD systems. The goal of this work is to provide this implementation and determine the feasibility of using *The Cloaker* model in practice. The primary concern of the authors when designing the security model of *The Cloaker* was that it adhere to the following design criteria:

1. The system should be safe and provide open access in emergencies.
2. Security and privacy needs to be maintained under adversarial conditions.
3. Battery life of the device needs to be maintained.

4. The system should respond to both the patient and the doctors within a reasonable and safe amount of time.

Modern IMD systems typically involve two devices: (1) the IMD device itself that is implanted in the patient, and (2) the Clinical Programmer that communicates to the IMD either to change settings or read important patient information from the IMD. In contrast, their system involves the addition of a third intermediate device. This third device, *The Cloaker*, acts as a proxy between the IMD and the Clinical Programmer to support the design goals that were outlined above. *The Cloaker* is foreseen as being a device very much like current wearable health devices in terms of scale and computational power and would be worn by the patient. In this model there are two use cases that need to be considered separately.

The first is when *The Cloaker* is present. When this is the case, all communication between the IMD and the Clinical Programmer must go through *The Cloaker*. There are several advantages to this setup. Since power consumption is a primary concern of IMD design, *The Cloaker* allows a means to use more computationally expensive cryptographic methods to pair with external Clinical Programmers without depleting the IMD device. The IMD and *The Cloaker* can be considered to be in a long-term relationship, so encrypted symmetric key protocols can be used to pair the two devices together and not attempt to pair with other devices. This shift of authenticating external devices through *The Cloaker* prevents Denial of Service attacks specifically targeting battery life through repeated authentication attempts.

The second use case to be considered is when *The Cloaker* is not present. As a means to fulfill their first design goal, the authors suggest that the IMD “fail open” in the

absence of *The Cloaker*. In an emergency situation this allows doctors to gain full access to the device by either removing or destroying the device. They suggest using biometric information such as the patient's pulse to indicate whether or not *The Cloaker* is present. This prevents against the possibility of *The Cloaker* being lost but still locking the device in the case of an emergency.

This work will provide an implementation of their proposed system called the BLE Cloaker that will use Bluetooth Low Energy (BLE) as the long-range wireless communication medium in conjunction with Out-of-Band (OOB) pairing through RFID to provide additional security. BLE is a good candidate for medical wireless transmissions due to its low power, compatibility with other sources of Electromagnetic Interference (EMI), and data transmission confidentiality [7]. If configured, BLE also supports 128-bit encryption as well as periodically changing the device address to a random value. There has been research that has shown the feasibility of using BLE directly in IMDs such as the BLE enabled Implantable Glucose Monitor [1]. There are also numerous examples of BLE being used as a means to connect wireless body sensors that take biometrics such as ECG to external devices including cell phones [10]. One reason that BLE is so appealing is that it has already been widely adopted by mobile phone developers, so in the future there is the potential for IMDs to directly use the patient's cellphone to upload medical information.

An RFID link will be used between the BLE Cloaker device and the IMD device to initiate the BLE pairing procedure using OOB. Although it won't be addressed in this implementation, the security of this link is also a concern and should be considered. There are several papers which outline methods for securing RFID links. Many of these

could be considered RFID versions of *The Cloaker* whose purpose is to act as an authenticated proxy between a resource constrained device and that of an external device attempting to read or update that constrained device. Some examples of these include the RFID Guardian which is a proposed device that is supposed to be kept near the person and block unauthorized access to a person's RFID enabled devices that are within range [8]. There has also been research that has shown that it is possible to implement RC5 on a resource constrained RFID microcontroller that performed relatively well in terms of both needed computational power and energy consumption [2]. It should be noted that all of the above research was geared towards developing cryptography methods for passive RFID tags whose goal was to keep the device as inexpensive as possible to make widespread manufacturing of these devices on household items economically feasible. Because IMDs are not produced in quite the ubiquitous scale as passive RFID tags, reducing cost in this manner is not a primary concern so it is reasonable to assume that there will be at least a marginal improvement in the amount of computational power available in IMDs for cryptographic purposes.

CHAPTER 3 – System Design

The overarching computing ecosystem of a patient using an IMD is complex and expands across multiple platforms and devices, as well as between multiple parties including doctors, clinicians, and software developers. The IMD directly connects to a Clinical Programmer or base station through either close proximity RF telemetry or long-range wireless techniques. In the case of a Clinical Programmer, a clinician or manufacturer technician will be present to update settings and take patient measurements. For home use, an IMD may connect to an external base station that will send patient data through the phone line to a backend database for future analysis by a doctor. In the future, it is possible that a mobile device using either a GSM or Wi-Fi network could replace this base station. Figure 1 shows what an example system might look like.

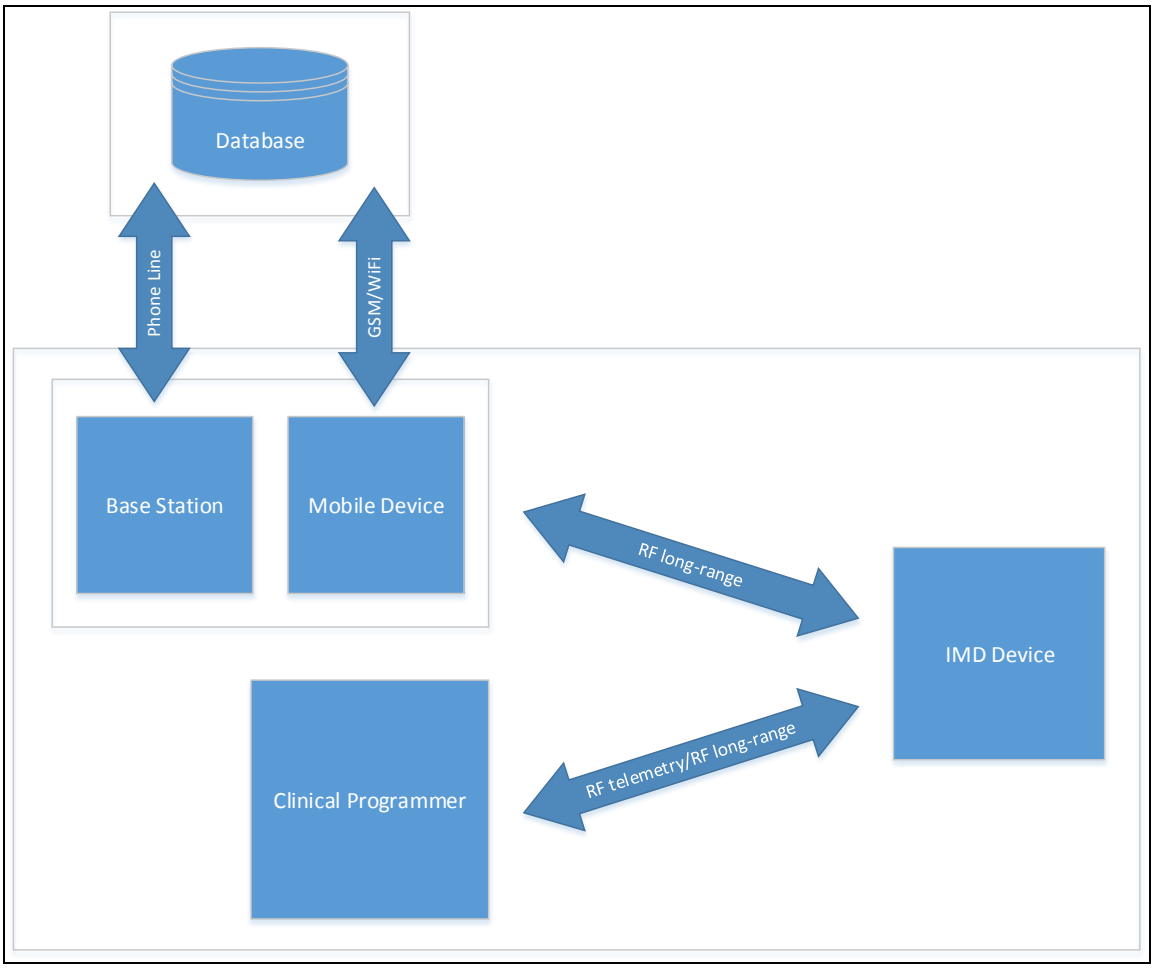

Figure 1: Typical IMD Ecosystem

3.1 System Overview

The security model presented in this work focuses on only a portion of the ecosystem outlined in Figure 1. Specifically it is concerned with the security of the direct link between the IMD Device and the external devices, both authorized and unauthorized, who wish to connect to it. This system has three parts: (1) the IMD device itself, (2) the BLE Cloaker, and (3) the external IMD programmer which could be any one of the three devices mentioned above, namely a Clinical Programmer, base station, or mobile device. Figure 2 presents a diagram of the various components of this system. Two different scenarios are outlined which were discussed in the Related Works section, one in which the BLE Cloaker is present and the other in which it is not. Each component will be discussed in more detail in the subsequent sections.

3.1.1 System Diagram

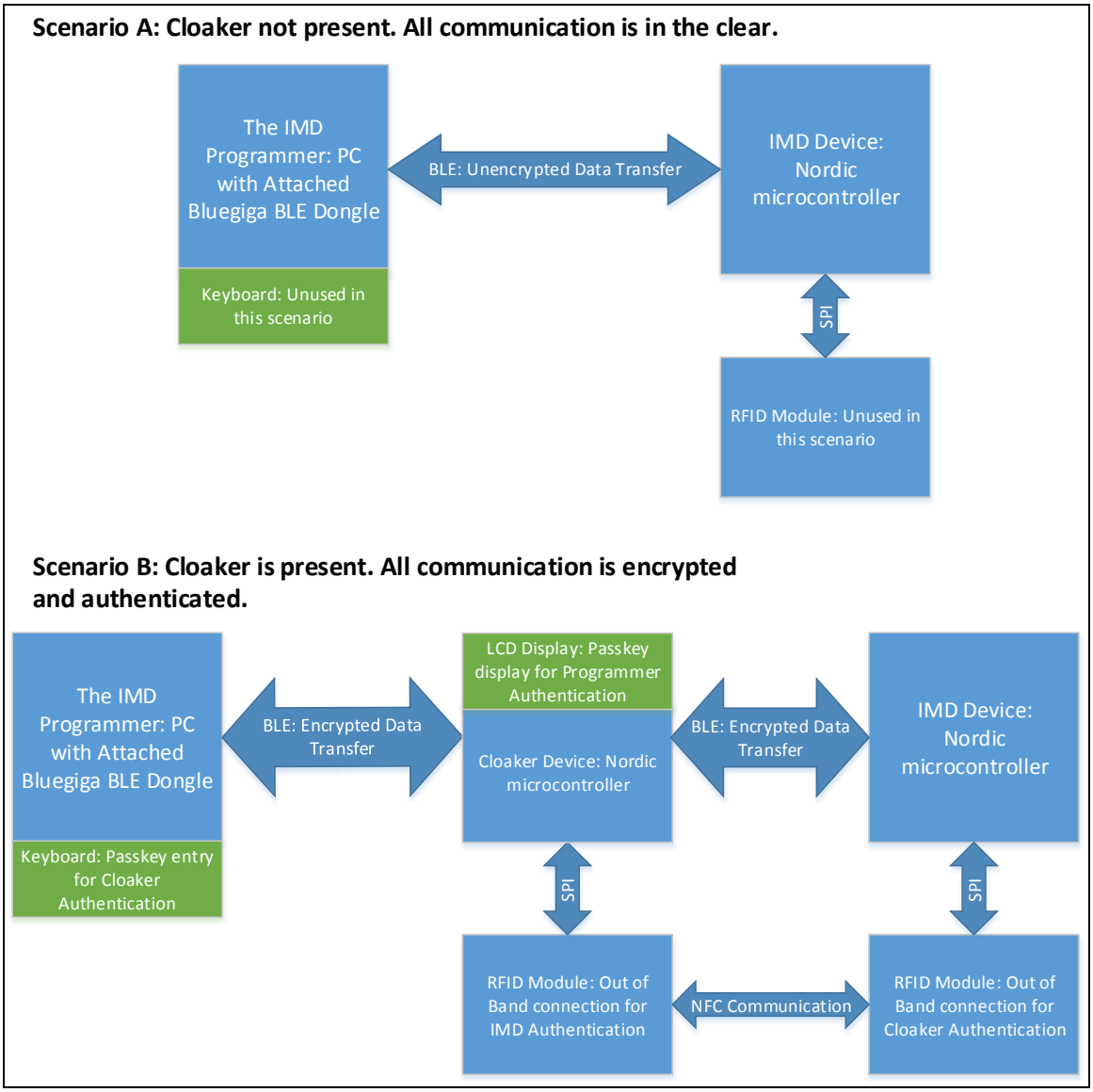

Figure 2: System Diagram outlining major components

3.1.2 BLE Overview

BLE follows a simple model that contains two different types of devices: a Central and one or more Peripheral devices. The Central device can also be referred to as the Master device, and is responsible for scanning for and connecting to Peripheral devices. Peripheral devices in a similar fashion are also known as Slave devices. They are responsible for advertising themselves and allowing Central devices to connect to them. Peripherals also have what are known as Services and Characteristics.

A Service is represented by a 128 bit Universally Unique Identifier (UUID) and is a way to create logical separations between the functionalities of a given Peripheral device. Each of these Services may have one or more Characteristic values that are also represented by 128 bit UUID's. Characteristics are a further way to separate different types of data within a Service. Each Characteristic has its own properties including whether it is allowed to be written or read to, as well as how many bytes of data a Characteristic can hold. An example of this is the Blood Pressure Service defined by the Blueooth Special Interest Group (Bluetooth SIG). This Service has several Characteristics including the "Blood Pressure Measurement" Characteristic and the "Intermediate Cuff Pressure" Characteristic. After a Central device connects to a Peripheral device, there is a standard method known as service discovery in which the Central discovers all of the Peripherals Services and Characteristics. Depending on the properties of each Characteristic, the Central will then be able to read and write to these Characteristics.

3.1.3 BLE Scatternet Topology

The Star topology is the most common and most supported configuration in which BLE is used today. This topology, described in the previous section, is where there is one master node that connects to one or more slave devices. In BLE the master is the only role which can initiate a connection request to other slave devices. Because of the inclusion of the BLE Cloaker node, what is known as a Scatternet topology will need to be used. The BLE Cloaker needs to be able to both connect to the IMD Device as well as be connected to by the IMD Programmer. A Scatternet topology allows the BLE Cloaker to act as both a master and a slave simultaneously, therefore making this dual connectivity requirement possible. Figure 3 shows what this Scatternet topology looks like in practice.

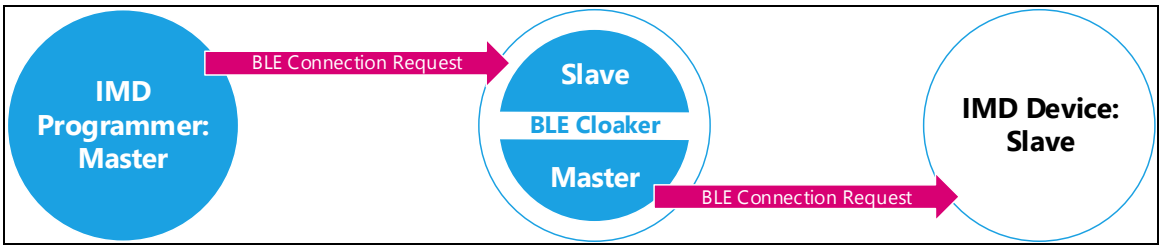

Figure 3: Diagram of the BLE Scatternet

3.2 System Components

Each of the three components, although related, have their own distinct responsibilities and roles within the system. The design of each of these components will be given in detail below. Software flow diagrams are provided for all three components outlining the order of code execution. Additionally, the IMD Device and the IMD Programmer were both written in C++ and as such have corresponding UML class diagrams. For technical reasons described later, the BLE Cloaker was written in C and a state diagram is provided instead. The BLE Programmer and the BLE Device components follow the State design pattern outlined in "Design Patterns: Elements of Reusable Object-Oriented Software", and the BLE Cloaker follows it to a lesser extent due to it not being written in an Object-Oriented language.

3.2.1 The IMD Device

In a real system the IMD Device would be responsible for taking patient measurements and providing necessary medical treatments. Since this is not a real IMD, dummy data will be sent across the BLE link. The IMD Device component is responsible for generating and providing this simulated data when it is requested by the IMD Programmer. The IMD Programmer will either request this information directly or indirectly through the BLE Cloaker depending on the current usage scenario. It is also responsible for responding to connection requests from the BLE Cloaker that are initiated by the RFID module as well as advertising itself openly to the IMD Programmer when the Cloaker is not present. See Figure 4 and Figure 5 respectively for the IMD Device's Software flow and UML diagrams. Note that the UML diagrams do not show all

functions prototypes as it would be impractical to do so in a diagram. They will be explained in the System Implementation section.

3.2.1.1 IMD Device Software Flow Diagram

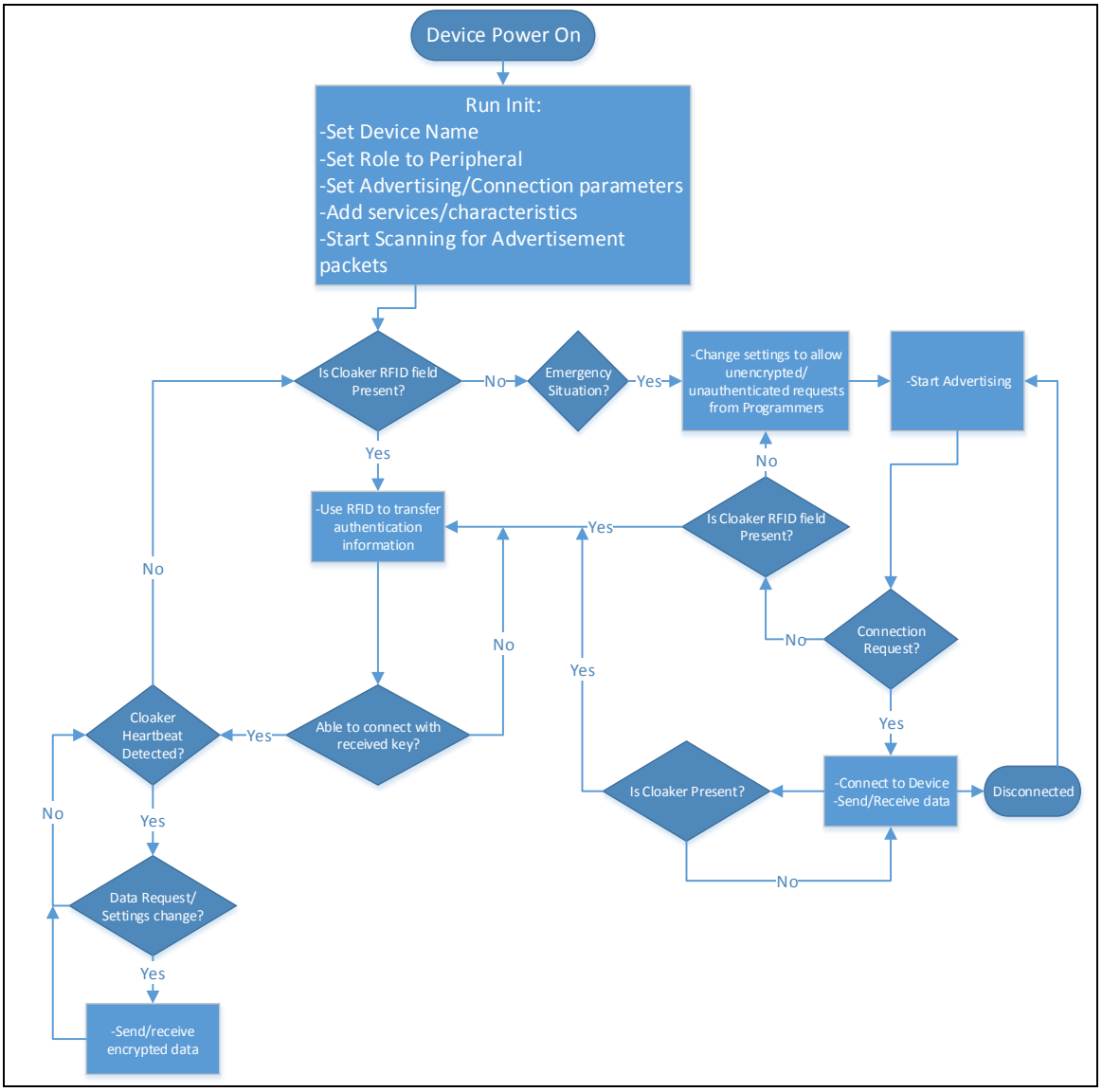

Figure 4: IMD Device Software flow diagram

3.2.1.2 IMD Device UML Diagram

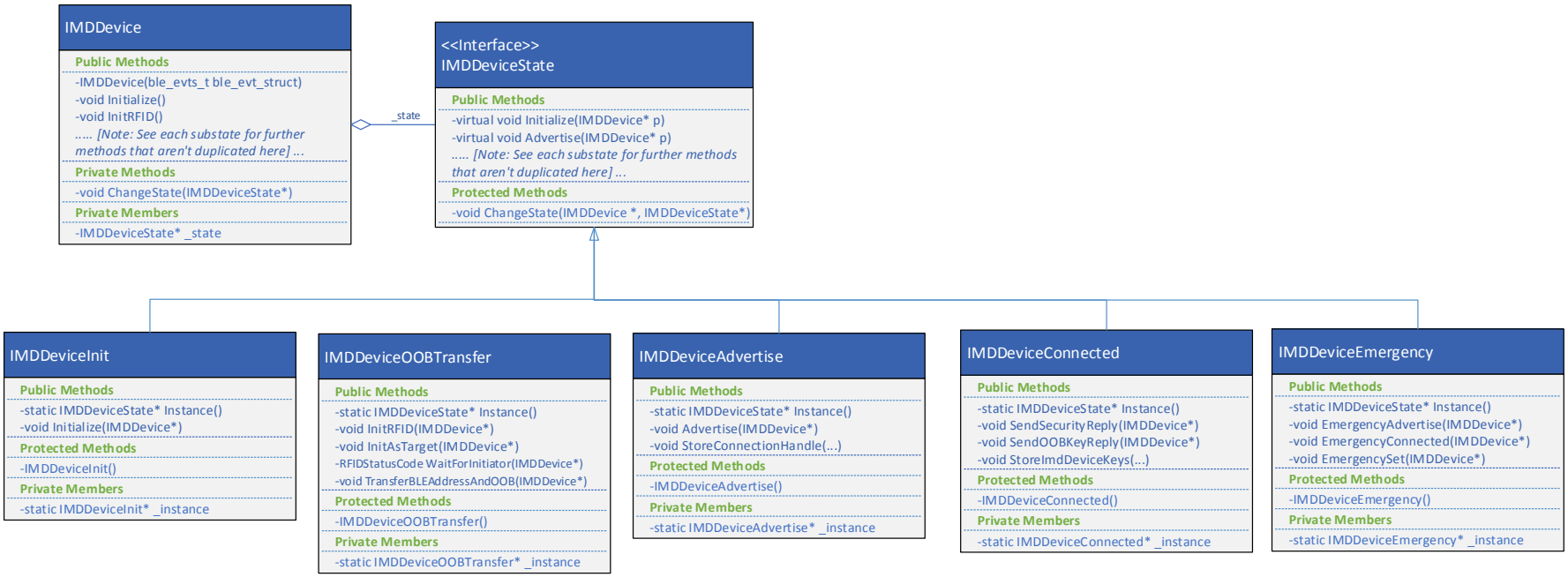

Figure 5: IMD Device UML Diagram

3.2.2 The BLE Cloaker

The role of the BLE Cloaker is to detect the RFID field from the IMD Device and subsequently transfer the appropriate keys and connect to it over BLE. Once connected, it is then responsible to advertise itself to the IMD Programmer and display its key on the attached LCD screen. From this point on it mediates the data transfer between the IMD Device and the IMD Programmer. If it loses the "heartrate" reading from the patient, it will disconnect from both the IMD Device and the IMD Programmer. See Figure 6 and Figure 7 for the BLE Cloaker's Software flow and State diagrams.

3.2.2.1 BLE Cloaker Software Flow Diagram

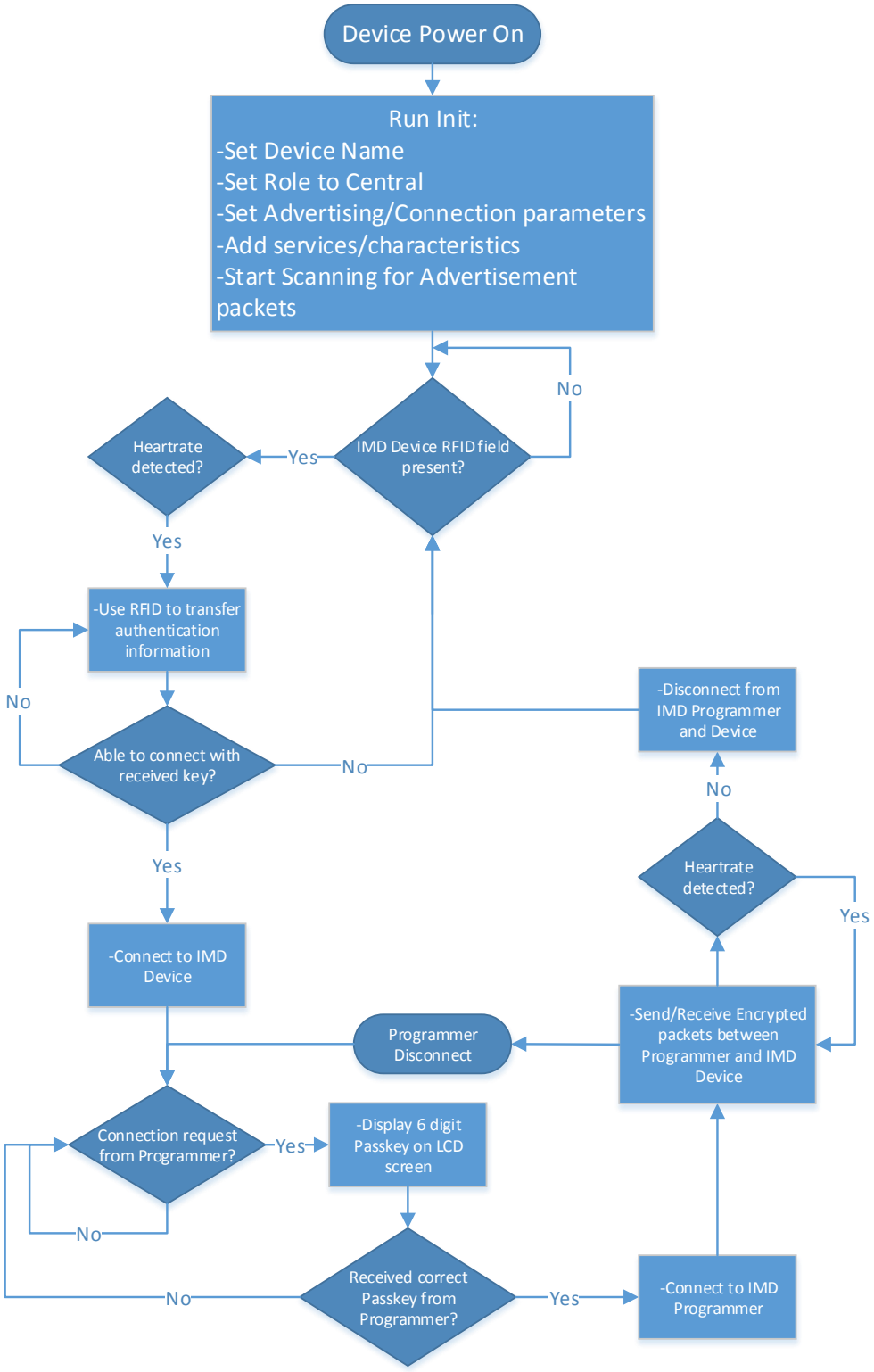

Figure 6: BLE Cloaker Software Diagram

3.2.2.2 BLE Cloaker State Diagram

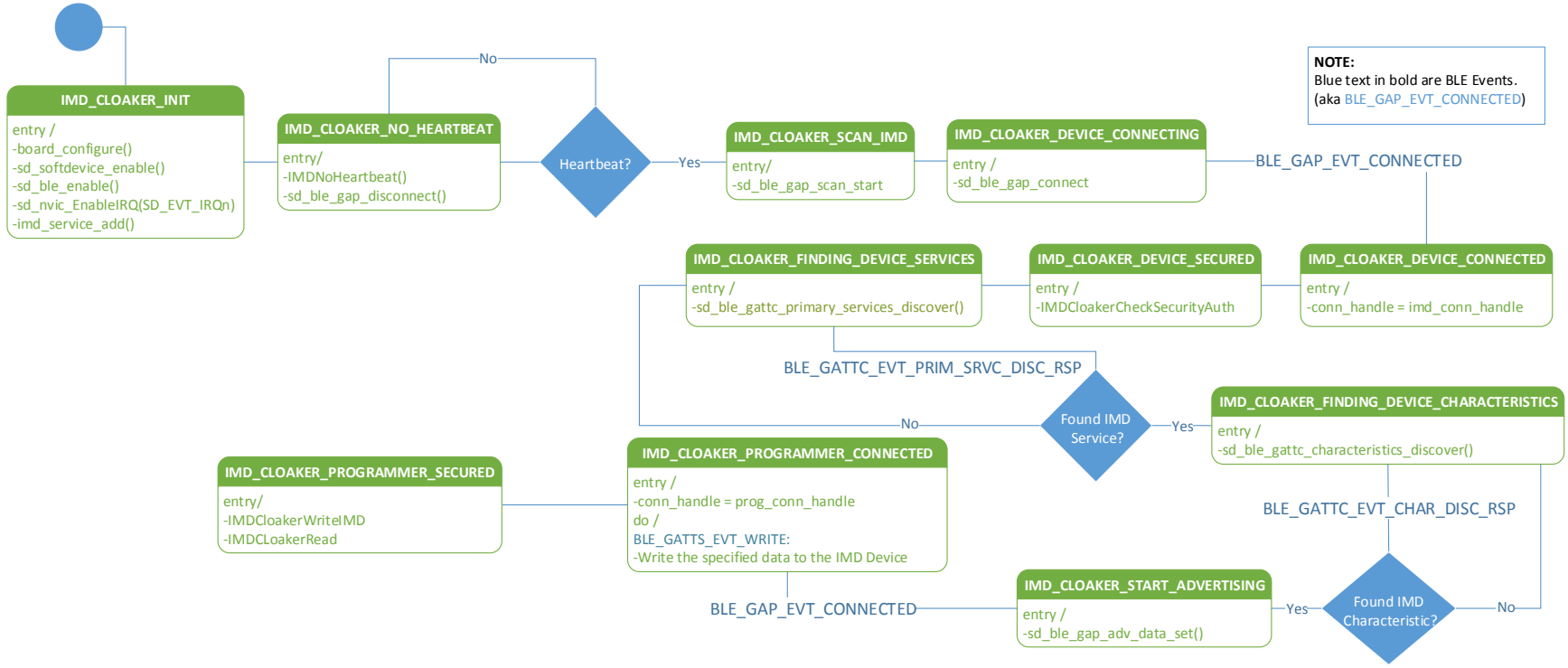

Figure 7: BLE Cloaker State Diagram

3.2.3 The IMD Programmer

The IMD Programmer needs to scan for the availability of both the BLE Cloaker and the IMD Device. The user, either a clinician or other authorized user, will then select which device to connect to. If the selection is the IMD Device and that device is in a state of emergency, it will connect without any security protocols used. If the BLE Cloaker is chosen, the IMD Programmer will send out a request to the BLE Cloaker to display its key on the LCD display. The user will then input this key into the IMD Programmer and the devices will be connected. After a connection is established to either device, settings can be updated and patient data can be received either directly from the IMD Device itself or indirectly through the BLE Cloaker's encrypted link again depending on the current use case. Figure 8 and Figure 9 show the Software flow and UML diagrams of the IMD Programmer.

3.2.3.1 IMD Programmer Software Flow Diagram

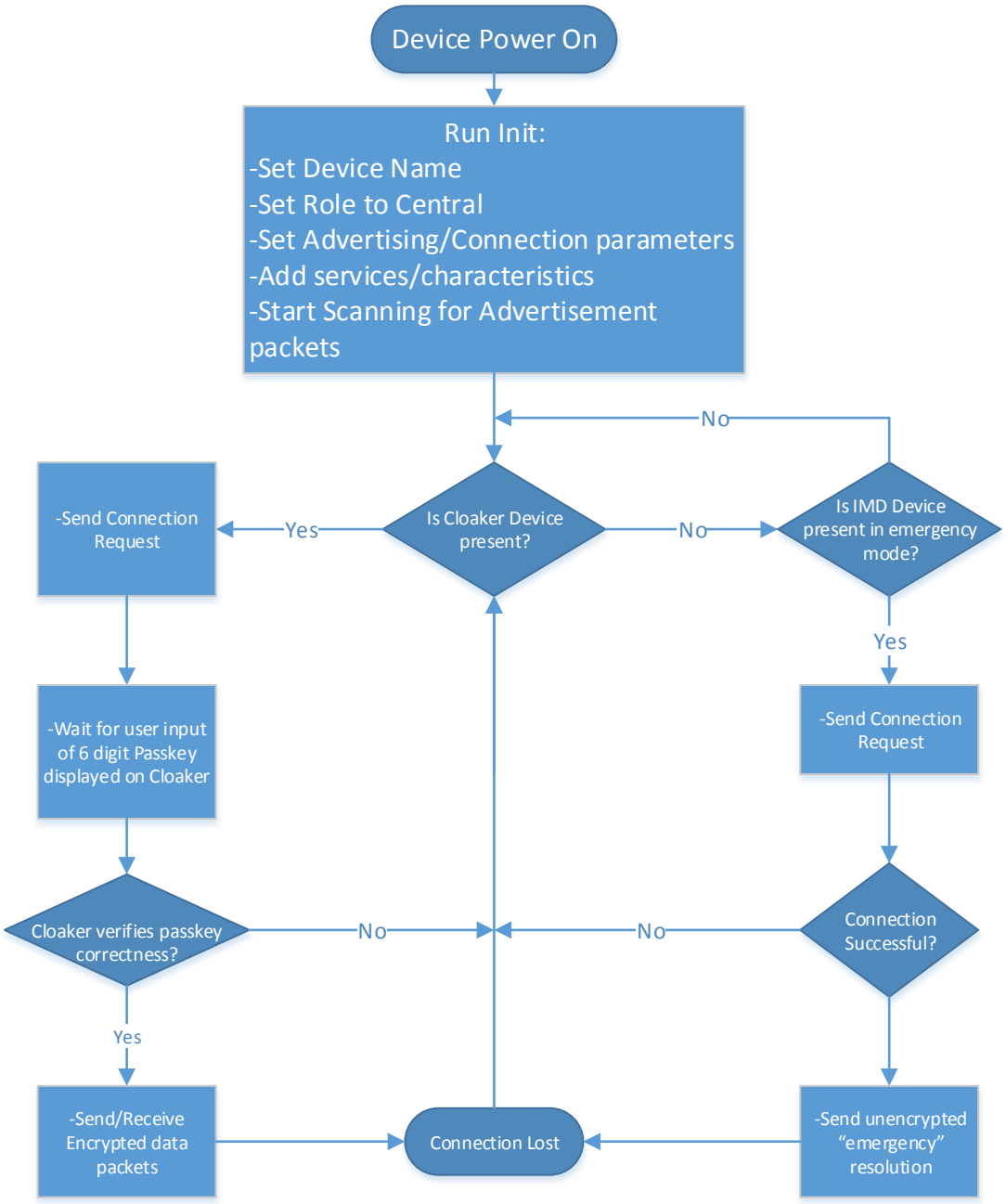

Figure 8: IMD Programmer Software Flow Diagram

3.2.3.2 IMD Programmer UML Diagram

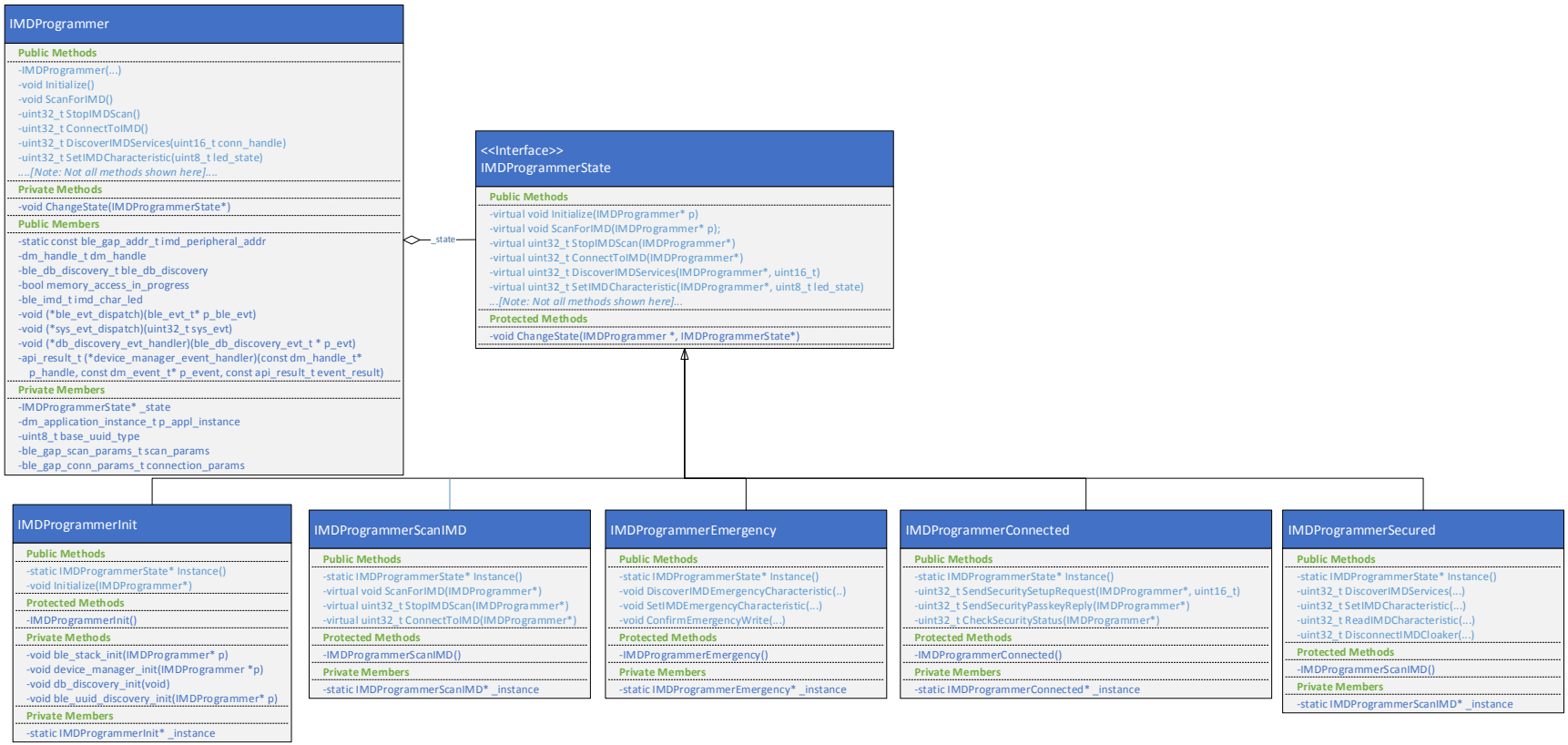

Figure 9: IMD Programmer UML Diagram

3.3 Security Model

In any software system that is hoping to implement security in some form or another, it is important to recognize the limitations of that system as well as define what type of adversaries it is designed to protect against. This is important for the purposes of both testing and practicality. The following sections will describe the particular types of adversaries that this protocol aims to protect against and what parts of the overall IMD ecosystem will be protected through the use of our protocol.

3.3.1 Adversary Threat Models

In general there are passive adversaries and active adversaries. Passive adversaries simply try to see what types of information they can obtain from a particular software system, whereas active adversaries, as the name suggests, actively try and manipulate the data they gather or try to exploit security vulnerabilities. Denial of Service (DoS) is another form of attack which attempts to deny a user or group of users' access to part or all of the functionality of a system. All of these attacks can be harmful and there are many different types of attacks that are possible. It should also be noted that it is assumed that any adversaries attacking this system are considered to be computationally bound, meaning that they do not have infinite resources in terms of computational power and time. The types of attacks that this protocol will provide protection against are detailed below.

3.3.1.1 Passive Adversaries

An adversary shall not be able to gain any useful information from passively eavesdropping on the BLE communication happening between the three components of

the system. Although it will be apparent that communication is happening, the adversary won't be able to tell the difference between what it is seeing and random data.

3.3.1.2 Active Adversaries

It will also protect against replay attacks in which the adversary attempts to replay back previous communications in the hopes that the receiving system will accept it as a new request. Authentication shall be provided, preventing an adversary from either performing the replay attacks noted above or other attempts at Spoofing the system using known plaintext or known ciphertext attacks.

3.3.1.3 Denial of Service

Adversaries will be prevented from carrying out DoS attacks that are aimed at draining the IMD battery life. The main method for doing this would be repeated attempts to authenticate with the IMD Device through the BLE connection procedure, thereby continually waking up the microprocessor and draining the battery. This will be prevented through the use of the BLE Cloaker. The IMD Device will not advertise itself and will only connect to the BLE Cloaker once the two devices have transferred keys through the RFID link.

3.3.2 BLE Link Security

There are two separate BLE links within this system that will both have different security schemes. Both BLE links use what is known as Secure Simple Pairing (SSP), however they each use a different method of passing their link keys. The first BLE link is between the IMD Device and the BLE Cloaker. These two devices will use what is known as Out-of-band (OOB) pairing to pass their link keys to each other through an RFID connection and finish the connection process. The second is the BLE link between

the BLE Cloaker and the IMD Programmer. This uses a passkey entry system in which the IMD Programmer requests the BLE Cloaker to display a 6-digit key to its LCD screen. The operator of the IMD Programmer then inputs this key. If the key matches, the two devices will successfully connect. Both of these pairing methods lower the likelihood of Man-in-the-middle (MITM) attacks as the keys are not directly transferred through BLE at any time under the two schemes, but instead rely on proximity and human interaction to transfer the keys.

3.3.3 Known Limitations

This work is primarily focused on the security of the BLE links between the devices as well as the general security of the system as it pertains to ensuring patient safety. As such, the security of the RFID link used for OOB pairing has not been enforced. This is not a trivial concern and needs to be addressed in future work. One common misconception is that the sheer proximity of RFID is enough to make it secure against both eavesdroppers and active adversaries attempting either passive or active attacks such as replay or MITM attacks. This is not true as it has been shown that by using relatively inexpensive equipment one can create an RF sniffer that can arbitrarily extend the communication distance of RFID devices.

CHAPTER 4 – System Implementation

This section outlines in detail how each portion of the BLE Cloaker model works in practice. This includes an in-depth overview of how BLE security has been implemented according to the BLE specification and what that looks like when using actual BLE hardware. In addition an explanation of how the IMD Device, BLE Cloaker, and the IMD Programmer along with their associated peripherals has been programmed will also be written up in this section.

4.1 BLE Security

As mentioned in the previous sections, the wireless communication link between the IMD and the outside world is the primary weak spot that an adversary can take advantage of if they wish to tamper with one of these devices. The wireless communication that is used in this implementation to connect the IMD to an external device is BLE. Aside from being significantly more energy efficient than previous versions of Bluetooth, one of the major design goals of BLE was to adhere to a higher standard of security. This work uses off-the-shelf BLE radios from Nordic Semiconductor in conjunction with some additional peripherals to implement the highest level of security that is possible using the BLE specification. This section gives an overview of how security is implemented in the BLE protocol.

4.1.1 Security Modes and Levels

The BLE specification outlines two different modes of security: mode one and mode two. Each of these modes has several levels of security associated with them. Security mode one has three different levels. Level one means that there is no security present at all, level two requires unauthenticated pairing with encryption thereafter, and

level three requires authenticated pairing with encryption. Mode two has two levels of security. Level one requires unauthenticated pairing with data signing and level two requires authenticated pairing with data signing. As security mode one level three requires both authentication during pairing as well as an encrypted connection thereafter, the National Institute of Standards and Technology (NIST) considers it to be the most secure option that the BLE standard provides [9]. This is the security mode and level that will be used in this project.

4.1.2 Pairing Phases

The BLE pairing process contains three distinct phases to create a secure connection between two different devices. In Phase 1, the Central device sends an initial pairing request and the Peripheral responds to this request. This is also when the Temporary Key (TK), which will be described later, is transferred between the two devices. In Phase 2, a Short Term Key (STK) is generated. The STK is generated using the TK that was transferred earlier along with randomly generated numbers. An encrypted link is temporarily started using the STK. Phase 3 is when the Long Term Key (LTK) is generated and exchanged between the BLE devices using the link that was encrypted using the STK. In addition to this, an Identity Resolving Key (IRK) as well as a Connection Signature Resolving Key (CSRK) may or may not also be generated and exchanged. Figure 10 shows the various steps involved in each phase.

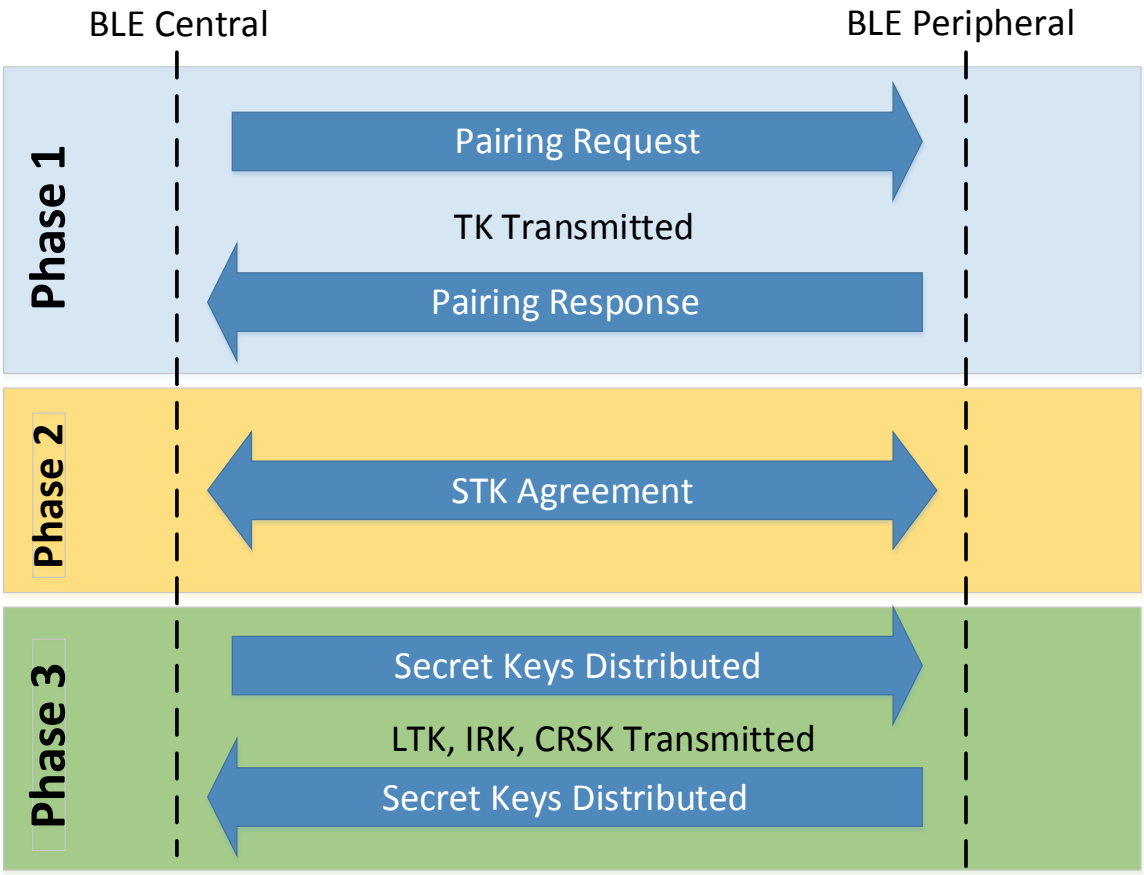

Figure 10: Phases 1 through 3 of the BLE pairing and encryption process

4.1.2.1 Pairing Phase 1

The primary purpose of pairing Phase 1 is to transfer the Temporary Key (TK). The TK is the primary component of what all of the keys that are created and transferred in the second and third phases are based off of. There are three different means of transferring the TK, and these variations are known as Secure Simple Pairing (SSP). All three of these variations are used at different points in the project.

The first variation for transferring the TK is known as Out of Band (OOB) pairing. As the name implies, a communications method that is different from BLE is used to transfer the TK between the Central and the Peripheral. This method provides protection against MITM attacks insofar as the OOB medium that is used is resistant to them. Near Field Communication (NFC), a close proximity version of RFID communication will be used in this project to perform OOB pairing as will be described later.

The second variation for transferring the TK is Passkey Entry. This is a common form of pairing that has been used by previous versions of Bluetooth. In order for it to work, at a minimum one device needs to have display capability and the other needs to have input capability such as through a keyboard. One device displays a randomly generated 6 digit numeric key and the other device types this displayed key in. If the key was input properly, the TK is generated from this 6 digit key and the devices move on to the second phase.

The final variation for transferring the TK is known as Just Works, and it should not be considered as an actual secure method of pairing. When Just Works pairing is used, the TK is set to all zeroes and the second and third phases carry on from there. This

is clearly not secure as the STK can easily be determined by an adversary since it is based off of the TK, which is known in this scenario. This model should not be used unless it is desirable to have an insecure link, which as will be explained does serve a purpose in the BLE Cloaker model.

4.1.2.2 Pairing Phase 2

After Phase 1 has completed and TK has been transferred, the process of generating and transferring the STK begins in Phase 2. There are several steps in this phase, the first of which is to authenticate each device using a verifier function to ensure that each device is using the same TK. This verifier function is defined in the BLE specification document and is known as the "Confirm value generation" function, or c1 for short. It uses 128-bit AES encryption under the hood and takes the following parameters:

- Mconfirm = c1(TK, Mrand, Pairing Request command, Pairing Response command, initiating device address type, initiating device address, responding device address type, responding device address)

Aside from TK and Mrand which are randomly generated the other parameters are static to each device. The exception is the Pairing Request and Response commands, which are always the same. Each device generates starts off the process by generating a 128 bit random number, known as Mrand in the case of the Central and Srand in the case of the Peripheral device. Then each device uses c1 to generate Mconfirm and Sconfirm respectively. The two devices exchange Mconfirm and Sconfirm, and then the Central transmits Mrand to the Peripheral. The Peripheral uses c1 to recalculate Mconfirm using Mrand, TK, and the other parameters. If the received Mconfirm and the generated

Mconfirm match, then the process continues with the Peripheral transmitting Srand and the central device recalculating Sconfirm. Assuming both Mconfirm and Sconfirm are successfully verified, each device uses the "Key generation function" s1 to generate the STK. This function takes the following parameters:

- STK = s1(TK, Srand, Mrand)

  This function also uses 128 bit AES encryption as its basis for key generation. At this point, the two devices use the STK to create a temporary encrypted link to transfer the remaining keys. Figure 11 shows this verification process more clearly.

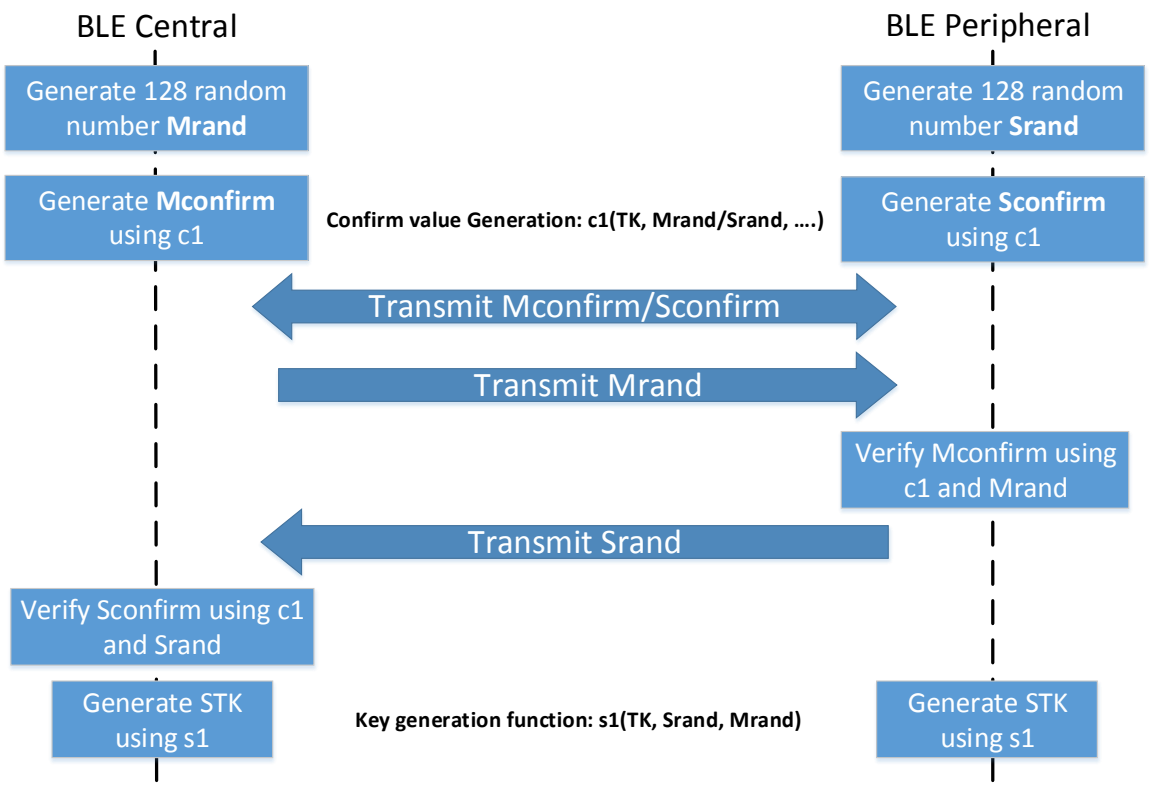

Figure 11: Diagram of the STK generation process in Phase 2

4.1.2.3 Pairing Phase 3

The third and final phase, Phase 3, is when the Long Term Key (LTK) is generated and distributed over the STK encrypted BLE link. The LTK is generated from yet another 128 bit AES based function known as the "Diversifying function" or d1. It has the following signature:

- LTK = d1(ER, DIV, 0)

ER is the Encryption Root which is a static and random 128 bit number. DIV is a 16 bit diversifier that is unique to each trusted device. In addition to the LTK, the IRK (Identity Resolving Key) and the CRSK (Connection Signature Resolving Key) can be generated in this phase. However for this application neither of these keys are necessary. The IRK is needed when the BLE privacy setting is enabled. When this feature is active, an advertising device will periodically change its publicly visible address so that it can't be followed by an unwanted adversary. In addition, this address can only be resolved using the IRK. This implies that devices need to have been previously bonded in order for this privacy feature to be used, which is not the case for this implementation. The CRSK is used for data signing, which is also not a feature that is being used in this project.

4.1.3 Message Encryption

The BLE specification makes use of the industry standard Advanced Encryption Standard-Counter with CBC-MAC (AES-CCM) algorithm for encryption. AES-CCM is a mode of operation for blocks of 128 bits in length. AES-CCM provides both authentication as well as encryption and uses an "authenticate-then-encrypt" scheme. It is used as would be expected to encrypt messages being sent over the BLE link, and the key to be used for AES-CCM is generated as follows:

- h4(LTK, KeyID, BD_ADDR_M, BD_ADDR_S) = HMAC-SHA-256(KeyID || BD_ADDR_M || BD_ADDR_S) / 2^{128}

KeyID is set to the string “btdk” which stands for “Bluetooth Device Key” and BD_ADDR_M and BD_ADDR_S are the Central and Peripheral addresses respectively.

4.2 IMD Services and Characteristics

For this work it was necessary to define two Services and three Characteristics to be used by each of the BLE components within the system. When creating “vendor specific” BLE Services and Characteristics rather than those that are predefined by the Bluetooth SIG, it is customary to use a common 128 bit base UUID for all of the necessary Services and Characteristics. The way this is done in the Nordic chipset is that the base UUID is given first, and then a different two byte value is then given for each Service and Characteristic that is to be used. When added, these two bytes take the place of bytes 12 and 13 of the base UUID. The 16 bit values used and their corresponding Services and Characteristics are outlined below. Each Service and Characteristic will be discussed in later sections:

- IMD_DEVICE_SERVICE_ENCRYPTED: 0x1701
    - IMD_DEVICE_CHARACTERISTIC: 0x1702
    - IMD_DEVICE_EMERGENCY_CHARACTERISTIC: 0x1703
- IMD_CLOAKER_SERVICE_ENCRYPTED: 0x1704
    - IMD_CLOAKER_CHARACTERISTIC: 0x1705

4.3 System Components

The picture in Figure 12 shows what the system looks like and highlights which microcontrollers and peripherals are a part of each of the three system components of the

BLE Cloaker model. The diagram in Figure 13 gives an overview of the communication protocols that each of these components uses to talk to their respective peripherals. This section will describe in detail how each of these components work both individually and together.

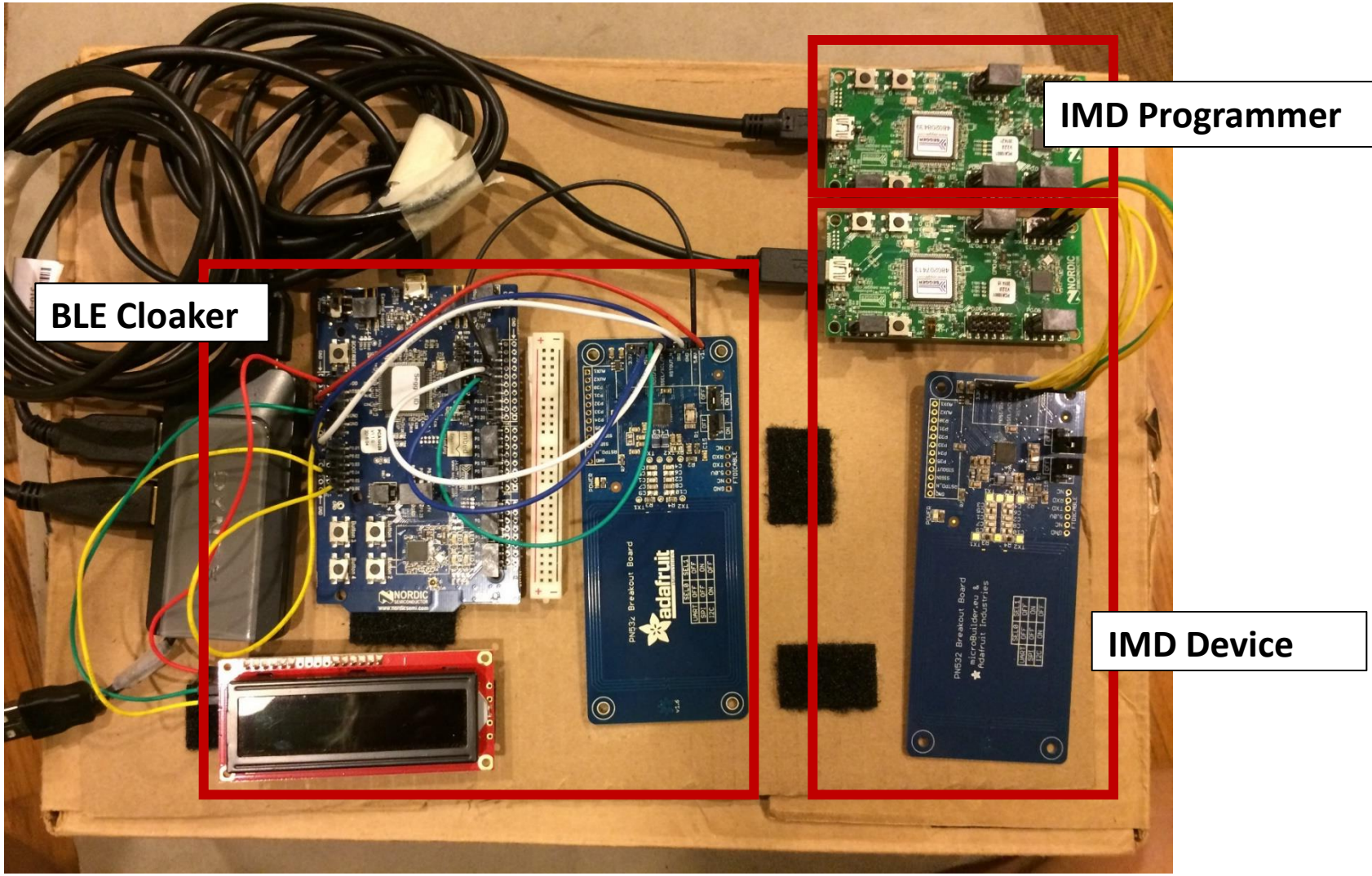

Figure 12: Picture showing an overview of the whole system

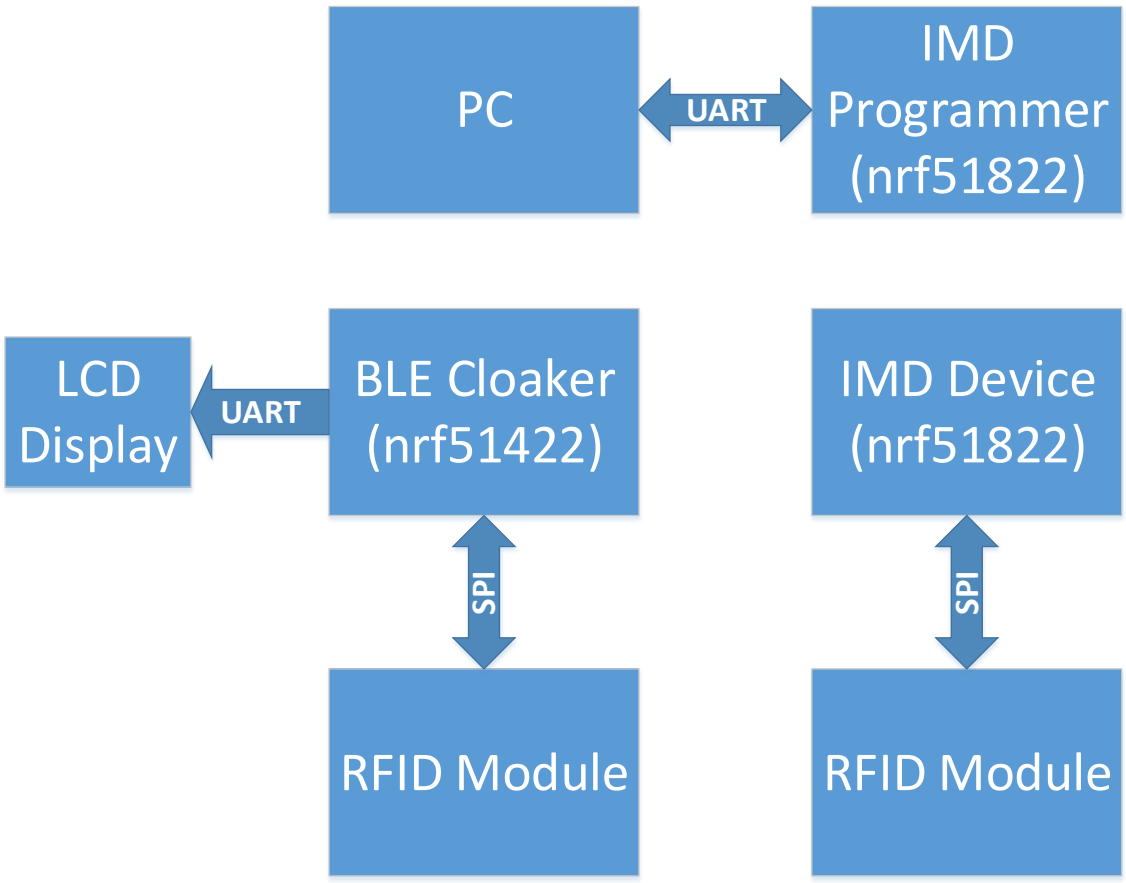

Figure 13: Overview of the various communication protocols used between the master and slave devices

4.3.1 RFID Slave Device

A pair of off-the-shelf NFC/RFID breakout boards designed by Adafruit were used as the means for transferring OOB data between the IMD Device and the BLE Cloaker. At the core of this board is the PN532 IC that acts as a slave device and supports SPI, I^2C, and UART commands. A SPI interface was used for both devices. These commands are used to setup the board for sending and receiving data between RFID enabled devices. Additionally there is an interrupt line available on the board so that the master microcontroller can be notified when new data has been received.

In addition to transferring OOB data, it is also used to transfer the BLE addresses of the IMD Device and BLE Cloaker back and forth. As a means to save power, the IMD Device doesn't start advertising until the RFID field of the BLE Cloaker's RFID device is within range and the OOB data has been transferred. The hardware within the PN532 IC takes care of the timing and protocol requirements of several different types of RFID protocols. The Near Field Communication Interface and Protocol (NFCIP-1) was used as it is one of the simplest schemes available. A transfer speed of 106 kbits/s was used as well as passive communication. Passive communication means that only the Initiator of the connection powers its RF field during communication, whereas the Target simply uses the power from the generated field for data transmission. By strategically making the IMD Device act as the Target, this allows for potential current savings.

After both the Initiator and the Target have been initialized, they wait until they are within range of each other. When this happens, further information is transferred and the Initiator decides on the baud rate and communication protocol to be used. After this, data can be repeatedly transferred back in forth. Figure 14 shows these steps.

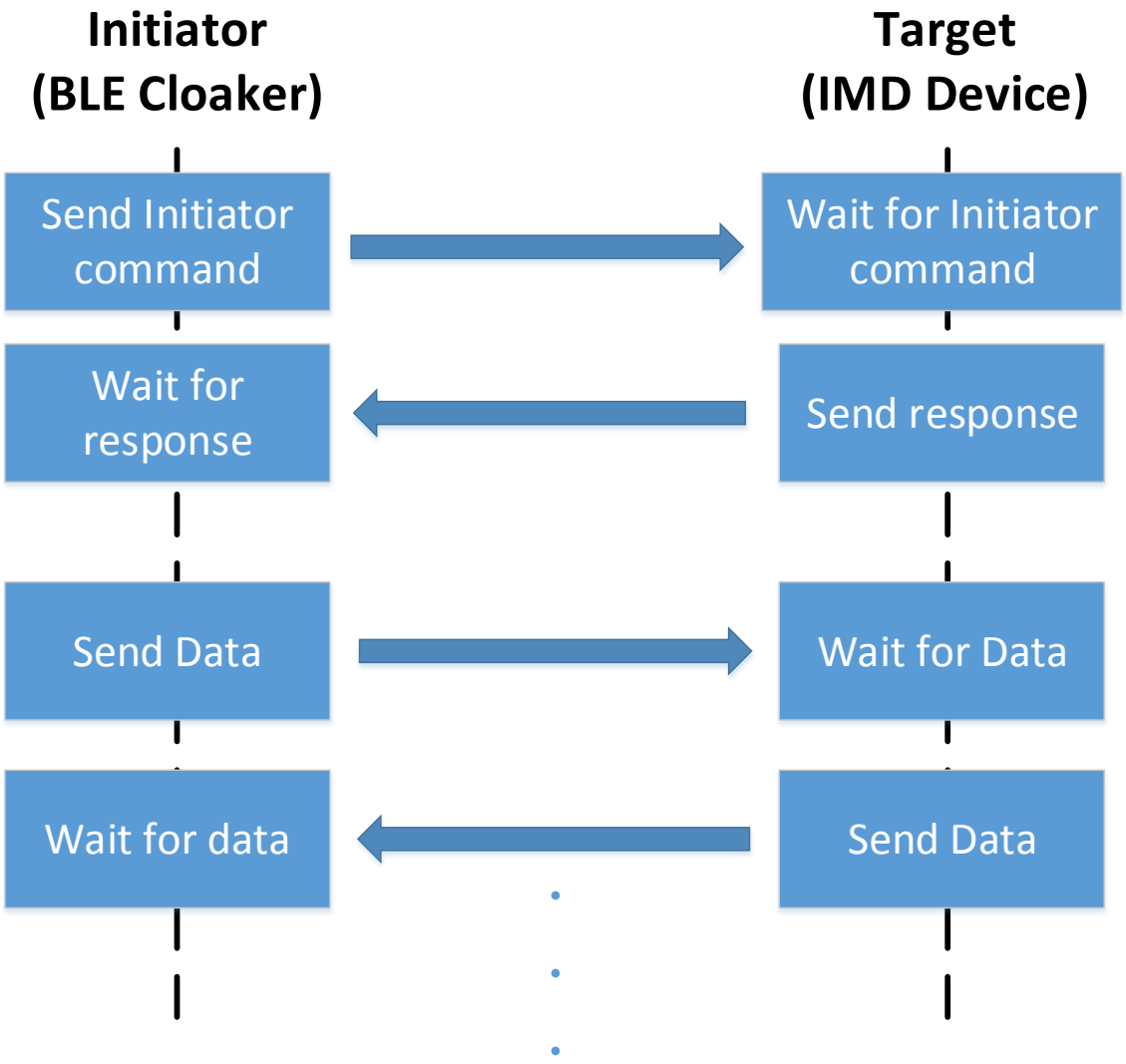

Figure 14: Diagram showing the necessary steps in an RFID setup and data transfer

4.3.2 IMD Device

The purpose of the IMD Device in this system is to emulate the wireless communication frontend of an actual IMD as closely as possible. There were two pieces of hardware that make up the IMD Device. The first is an NRF51822 IC on the mbed kit, a breakout board manufactured by Nordic Semiconductor. This is the primary device and contains a standalone processor as well as a BLE radio. Attached to it through a SPI connection is one of the two RFID slave devices for receiving OOB data from the BLE Cloaker. The software is split into two distinct sections on the flash memory: the Nordic S110 SoftDevice and the application code. The S110 SoftDevice is Nordic's proprietary Peripheral BLE stack that handles all BLE related operations as required by the BLE specification. The application code makes function calls into the SoftDevice to control the BLE radio and to carry out the IMD Devices two primary functions.

In its idle state, the IMD Device makes a call to *sd_app_evt_wait()*. This function call puts the IC into a low power state until either a BLE or other event happens. The IMD Device has two modes. The normal case is it enters into an encrypted connection with the BLE Cloaker. The other case is when it enters into emergency mode and can enter into an unencrypted connection with anyone who is scanning for it. The circumstances that cause both of these modes to occur as well as a detailed description of these modes is described below. Prior to either of these modes occurring however, the IMD Device enters the *IMDDeviceInit* state. In this state the IMD Device initializes the S110 BLE stack, sets up the Device Manager, and sets up the General Access Protocol (GAP) settings required for a BLE connection to occur. The Device Manager is responsible for such tasks as intercepting incoming BLE events and managing peer

connections and their corresponding security keys. Additionally it adds the IMD Device encrypted BLE Service and both the encrypted Characteristic for normal operation and the unencrypted Characteristic to be used in emergency mode. Finally it sets the connection security parameters to require bonding, MITM protection, and states that it supports OOB and has no I/O capabilities.

4.3.2.1 Normal Operation with the BLE Cloaker

The IMD Device enters into normal operation when the RFID field of the BLE Cloaker is detected. When this happens, the IMD Device enters the *IMDDeviceOOBTransfer* state and the OOB data as well as the BLE address of itself and the BLE Cloaker are transferred. For security purposes, these BLE addresses are set to random values every time a new connection is formed between the IMD Device and the BLE Cloaker. This helps to ensure that an adversary is not able to continually have access to a device even if a successful attempt was made in the past.

After the OOB transfer completes, the IMD Device goes into the *IMDDeviceAdvertise* state. The advertising settings are set to connectable directed. What this means is that the IMD Device will only attempt to connect one device, namely the BLE Cloaker whose BLE address was transferred during the OOB link. This mode allows for a very fast connection with a 1.28 second advertising period. This short amount of advertising helps to reduce power consumption.

Once connected, the IMD Device enters the *IMDDeviceConnected* state, at which time it replies to a request made by the BLE Cloaker regarding what its BLE security requirements and abilities are. In this case, OOB security with MITM protection is required so the IMD Device responds with the TK it received over the OOB connection.

The second and third phases of the pairing sequence are done by the SoftDevice without the need for user intervention. Once this process completes, the LTK is stored and encrypted reads and writes from the IMD encrypted Characteristic are now possible.

4.3.2.2 Emergency mode operation without the BLE Cloaker

Prior to being connected to the BLE Cloaker, an IMD emergency can be simulated through a button press on the board. When this happens, the IMD Device goes into the *IMDDeviceEmergency* state and begins to advertise using the connectable and undirected settings. This means that any device who is listening will be able to connect to the IMD Device. In this scenario, the unencrypted IMD emergency Characteristic is added to the advertising data. The reason for this is so that the IMD Programmer will be able to parse the advertising packet for the emergency Characteristic UUID and thereby be able to find IMD Devices in emergency mode. If a connection request comes in from the IMD Programmer, the IMD Device will immediately connect using the Just Works pairing scheme. From here, the IMD Programmer can write to the unencrypted emergency Characteristic and resolve the "emergency". Something of note is that although no security is required to access the emergency Characteristic, the application does need to internally authorize access to it. This is so that in normal operation, although the emergency Characteristic is visible to the Central device, the application can refuse it access to reading or writing to it inappropriately. The motivation for this lack of security is that during a life threatening emergency, it would be better to have the IMD fail open and be accessible so that a medical practitioner may be able to help the patient more efficiently.

4.3.3 BLE Cloaker

The purpose of the BLE Cloaker is to act as an intermediary communications device between the IMD Device and the IMD Programmer. The BLE Cloaker consists of three distinct pieces of hardware. The NRF51422 IC broken out onto the nrf51 Development Kit by Nordic Semiconductor is again the master and commander of this component. Attached to it as slave devices are the RFID module for OOB transfers, and an LCD display for both status messages as well as displaying the passkey for securely pairing with the IMD Programmer.

The Nordic S130 SoftDevice was used in conjunction with user application code. The S130 SoftDevice is unique because it allows a single device to act as the Central and the Peripheral simultaneously. This was necessary as the BLE Cloaker needs to be able to scan for and connect to the IMD Device as a Central and then be able to advertise and be connected to by the IMD Programmer as a Peripheral. The downside to using the S130 SoftDevice is that up until recently it was only available as an Alpha build, meaning that it was somewhat difficult to use and there was only a small amount of example code available. Another difficulty is that the S130 has only been tested to run on version 3 boards and for some reason does not appear to play nicely with C++, which was why C was used for coding the BLE Cloaker instead. At any given time, the BLE Cloaker is in one of three overarching modes of operation. These are described below.

4.3.3.1 No devices connected

Before any devices are connected, the BLE Cloaker runs through its initialization routines and sets up the important BLE parameters in a similar fashion as the IMD Device. In addition it adds the BLE Cloaker Service and Characteristic for later discover

by the IMD Programmer. One feature of note pointed out in the Cloaker model is the presence of a biometric indicator that determines whether or not the BLE Cloaker is attached to a person or not. When this biometric indicates that the BLE Cloaker isn't attached to the patient anymore, it immediately disconnects itself from the IMD Device. The reason for this is so that if an emergency situation occurs, doctors can simply remove the BLE Cloaker to gain emergency access to the IMD Device as it fails open in the absence of the BLE Cloaker. To simulate this biometric reading, a simple GPIO line was added. If it is plugged in, it means that the BLE Cloaker is attached to a person and can be connected to the IMD Device. Otherwise, no connections are possible and any current connections will be severed.

Assuming the presence of the simulated biometric indicator, the BLE Cloaker will exit this mode once it detects the RF field of the IMD Device and will transfer its OOB data and its randomly generated BLE address. The BLE Cloaker actually generates the OOB data and its random BLE address using a built in thermal noise random number generator. After the IMD Device's BLE address has been transferred, the BLE Cloaker attempts to connect to it and sends the required security requests. The pairing sequence ends with the reception of the LTK.

4.3.3.2 IMD Device connected

Now that a secure connection has been formed with the IMD Device, the BLE Cloaker will use the LCD screen to prompt the user if they would like to start advertising. If they accept by pressing a button, the BLE Cloaker begins advertising and will display its BLE address on the LCD. This address will be entered in by the user of the IMD Programmer to ensure a proper connection is made. Once connected to the IMD

Programmer, the LCD will display a 6 digit key. If this key is entered correctly by the IMD Programmer, the LTK is transferred and the connection is secured.

4.3.3.3 IMD Programmer and IMD Device connected

Now that the three devices have been connected together, the BLE Cloaker can act as a data proxy between the IMD Programmer and the IMD Device. In the case of a data write, the IMD Programmer sends an encrypted write request to the BLE Cloaker who then sends another encrypted write request to the IMD Device. Performing an encrypted read is slightly more complicated. The reason for this is that because the IMD Programmer initiates the read, the BLE Cloaker needs to first read the value from the IMD Device before it replies to the IMD Programmer. To accomplish this, the BLE Cloaker Characteristic was setup to require authorization before being read. This way the IMD Programmer is expecting an intermediary reply to tell it if it was granted access or not. Before this reply is sent, the BLE Cloaker sends the read request to the IMD Device and gets the value. Then it authorizes the IMD Programmer's read request and sends back the most current data.

4.3.4 IMD Programmer

The purpose of the IMD Programmer is to emulate the functions of a Clinical Programmer by sending commands and receiving responses back from the IMD Device either directly when it is in emergency mode or indirectly when connected to it through the BLE Cloaker. Like the IMD Device, the brain of the IMD Programmer is an NRF51822 IC and corresponding breakout board. It is also connected to a PC through a UART connection to display commands and gather user input through the terminal. It is using the Nordic S120 SoftDevice, which is their proprietary Central BLE stack. The

IMD Programmer is either scanning for or connected to a BLE Cloaker, or scanning for or connected to an IMD Device in emergency mode.

4.3.4.1 Scanning for the BLE Cloaker

In this mode the IMD Programmer first waits for the user to input the BLE address of the BLE Cloaker they wish to connect to. After the address is inputted correctly, the user is prompted to enter the 6 digit security code displayed on the LCD screen of the BLE Cloaker. After this number is inputted, the IMD Programmer and the BLE Cloaker are in a secure connection. The user will now be prompted to either write or read from the IMD Device through the BLE Cloaker, or disconnect from it altogether.

4.3.4.2 Scanning for the IMD Device in emergency mode

When scanning for IMD Devices, the IMD Programmer will asynchronously intercept any incoming advertising packets from BLE devices in close proximity. However it won't connect to any of these devices unless their advertising packet contains the UUID of the IMD emergency Characteristic. Once the IMD Device has been found using this method, the IMD Programmer connects to it. The user is then immediately asked if they wish to resolve the emergency. After the user accepts, the IMD emergency Characteristic is reset and this triggers the IMD Device to disconnect from the IMD Programmer to prevent further unsecured access.

CHAPTER 5 – Issues, Assumptions, and Limitations

5.1 Issues and Assumptions

5.1.1 BLE Issues and Assumptions

The general communication scheme for Bluetooth Low Energy (BLE) devices is a Star topology in which a single master is connected to one or more slave devices. For this work however, it is necessary to use what is known as a Scatternet topology. Previous versions of the Bluetooth specification supported this topology, but the BLE specification has deemed this an atypical use case and as such it is not widely supported by BLE device manufacturers. Because of this fact Nordic, the manufacturer of the BLE ICs that will be used, has only recently come out with a production version of their software stack that supports the Scatternet topology. As it is such a premature version of the code, there are various bugs that remain to be worked out that make it difficult to use.

5.1.2 RFID Issues and Assumptions

Modern IMDs are equipped with RF induction technology that allows them to communicate with external Clinical Device Programmers. These programmers require the use of a strong inductive wand in order to be able to penetrate the tissue of the human body and communicate reliably with the IMD. The RFID microcontrollers used in this work were designed to communicate with passive RFID cards and mobile devices. As such the inductive field generated is not strong enough to communicate with an implanted IMD. This work is meant as a proof of concept with the assumption that future work would overcome this problem. An obvious solution would be to manufacture a device that generated a field that was indeed strong enough to penetrate human tissue and communicate with an IMD. Another possible scenario to explore is the notion that future

IMDs may be equipped with some form of subcutaneous RFID technology that would not require as strong of an inductive field as in past devices.

5.2 Current Limitations and Future work

5.2.1 BLE 4.1

The Nordic BLE IC's used in this project only support version 4.1 of the BLE standard. This version of BLE has known security issues, mainly the lack of protection against passive eavesdroppers. Pairing phases one and two are both done in the clear, meaning that it is possible for adversaries to listen in and obtain the TK and random numbers during this time frame. This information would make it trivial to determine the STK and use that to obtain the LTK. Future work would be to upgrade the current system to a BLE IC that has support for version 4.2 of the BLE standard. This version of BLE is both FIPS and NIST compliant, and it uses Elliptical Curve Diffie-Hellman (ECDH) public key cryptography to provide protection against passive eavesdroppers.

5.2.2 NFC/RFID Encryption

The current means of transferring OOB data using NFC is not secure against MITM attacks. Currently no methods of securing and encrypting this data as it is transmitted have been implemented. Future work would be to research and implement an appropriate encryption scheme.

5.2.3 IMD Device Emulation

Both the Nordic IC as well as the RFID module draw too much current to be feasibly used within an IMD of any kind. The RFID module itself draws 20mA of current after it has been initialized, which is much too high for this application. Future work will involve researching and selecting more power efficient devices.

CHAPTER 6 – Key Contributions and Conclusions

This work furthered the research of Kevin Fu et al. by implementing and looking into the feasibility of their Cloaker model using actual hardware and the ubiquitous BLE protocol as the wireless medium. This work also points out both the strengths and weaknesses of using BLE 4.1 as a wireless protocol for IMD's. It also provides a good baseline as well as suggestions as to where future researchers should focus their efforts in this new and emerging field of medical device security.

REFERENCES

1. Ali, Mai, Lutfi Albasha, and Hasan Al-Nashash. "A Bluetooth low energy implantable glucose monitoring system." Microwave Conference (EuMC), 2011 41st European. IEEE, 2011.
2. Chae, Hee-Jin, et al. "Maximalist cryptography and computation on the WISP UHF RFID tag." Wirelessly Powered Sensor Networks and Computational RFID. Springer New York, 2013. 175-187.
3. Denning, Tamara, Kevin Fu, and Tadayoshi Kohno. "Absence Makes the Heart Grow Fonder: New Directions for Implantable Medical Device Security." HotSec. 2008.
4. Greatbotch, Wdson, and Curtis F. Holmes. "History of implantable devices." (1991).
5. Halperin, Daniel, et al. "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses." Security and Privacy, 2008. SP 2008. IEEE Symposium on. IEEE, 2008.
6. Halperin, Daniel, et al. "Security and privacy for implantable medical devices." Pervasive Computing, IEEE 7.1 (2008): 30-39.
7. Omre, Alf Helge, and Steven Keeping. "Bluetooth low energy: wireless connectivity for medical monitoring." Journal of diabetes science and technology 4.2 (2010): 457-463.
8. Rieback, Melanie R., Bruno Crispo, and Andrew S. Tanenbaum. "RFID Guardian: A battery-powered mobile device for RFID privacy management." Information Security and Privacy. Springer Berlin Heidelberg, 2005.
9. Scarfone, Karen, and John Padgette. "Guide to bluetooth security." NIST Special Publication 800 (2008): 121.

10. Yu, Bin, Lisheng Xu, and Yongxu Li. "Bluetooth low energy (BLE) based mobile electrocardiogram monitoring system." Information and Automation (ICIA), 2012 International Conference on. IEEE, 2012.