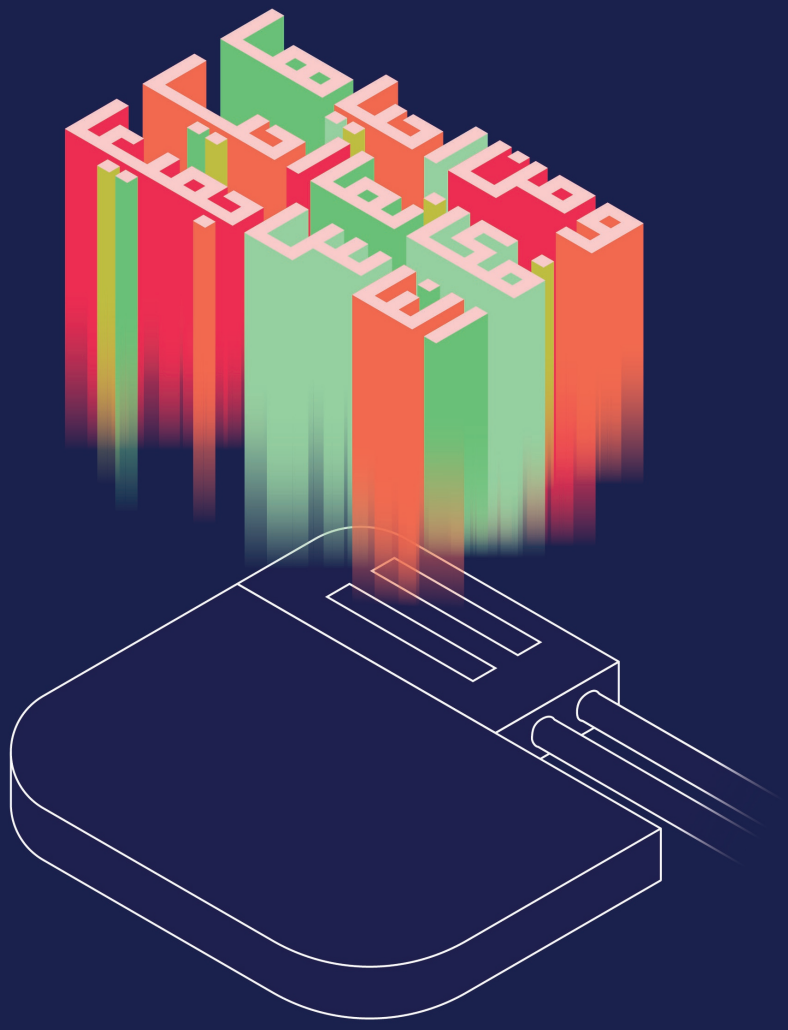
On the Security and Privacy of Implantable Medical Devices



Muhammad Ali Siddiqi

On the Security and Privacy of Implantable Medical Devices

Muhammad Ali Siddiqi

On the Security and Privacy of Implantable Medical Devices

Beveiliging en privacy van implanteerbare medische apparaten

Muhammad Ali Siddiqi

Keywords: Implantable medical device, security, denial-of-service attack, battery-depletion attack, energy harvesting, zero-power defense, security protocol, authentication, non-repudiation, emergency access, body-coupled communication, ultrasound, technical debt, embedded software, smart card

Layout: Muhammad Ali Siddiqi & Robert M. Seepers

Cover: Muhammad Ali Siddiqi & Mercedes Benjaminse
The cover shows an implantable medical device under an excerpt from the Qur'anic verse (5:32): "… and whoever saves a life, it will be as if they saved all of humanity."

Printed by: ProefschriftMaken BV

ISBN: 978-94-6423-357-5

On the Security and Privacy of Implantable Medical Devices

Beveiliging en privacy van implanteerbare medische apparaten

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
rector magnificus

Prof.dr. F.A. van der Duijn Schouten

and in accordance with the decision of the Doctorate Board.
The public defence shall be held on

Wednesday 29 September 2021 at 1300 hrs
by

Muhammad Ali SIDDIQI
born in Hyderabad, Pakistan

Erasmus University Rotterdam

Doctoral Committee

Promotor:

Prof.dr. C.I. de Zeeuw

Other members:

Prof.dr.ir. A.F.W. van der Steen
Prof.dr.ir. W.A. Serdijn
Prof.dr.ir. N. Mentens

Copromotor:

Dr.ir. C. Strydis

Dedicated to Mama, Baba, Omer, Osman and my wife Farwa

Contents

List of Tables

List of Figures

List of Acronyms

ADC	Analog-to-Digital Converter
ANS	Answer
APT	Acoustic Power Transfer
ASK	Amplitude-Shift Keying
AVISPA	Automated Validation of Internet Security Protocols & Applications
BCC	Body-Coupled Communication
CMAC	Cipher-based MAC
CMD	Command
CPP	Client-Puzzle Protocol
DCB	Dynamic Cardiac Biometric
DH	Diffie–Hellman (key exchange)
DoS	Denial of Service
ECC	Error Correcting Codes
ECG	Electrocardiography
ECoG	Electrocorticography
EH	Energy Harvesting
EMV	Europay, Mastercard, and Visa (payment method)
FCC	U.S. Federal Communications Commission
FDA	U.S. Food and Drug Administration
GPIO	General-Purpose Input/Output
HW	Hardware
IC	Inductive Coupling
IEEE	Institute of Electrical and Electronics Engineers
IMD	Implantable Medical Device
IoT	Internet of Things
IPI	Inter-Pulse Interval
IPT	Inductive Power Transfer
MAC	Message Authentication Code
MCU	Microcontroller Unit
MITM	Man in the middle (attack)
OOB	Out Of Band
OTA	Over The Air
PWM	Pulse-Width Modulation
PZT	Lead-Zirconate-Titanate (transducer)
RF	Radio Frequency
RFID	Radio-Frequency Identification
RFPT	RF Power Transfer
RISC	Reduced Instruction Set Computer
rPPG	Remote Photoplethysmography

RTC	Real-Time Clock
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SW	Software
TD	Technical Debt
TRX	Transceiver
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
WBAN	Wireless Body Area Network
WPT	Wireless Power Transfer
ZPD	Zero-Power Defense

CHAPTER 1

1

Introduction

Over the years, medical devices for personalized therapy using electrical stimulation have evolved from cumbersome huge devices, such as the first artificial pacemaker from 1932, shown in Figure 1.1, to much smaller ones, such as *implantable* pacemakers, cardioverter defibrillators, neurostimulators etc.; see Figure 1.2. These devices, which are collectively known as Implantable Medical Devices (IMDs), have come a long way starting with the introduction of the first implantable pacemaker in 1958.

The early devices, such as the one from Chardack and Greatbatch in Figure 1.3, allowed manual programming of the device parameters, such as the electrical output and pulse-repetition rate. A Keith needle was employed to percutaneously access a potentiometer within the implant using an opening on the device. Turning it clockwise or anticlockwise changed the setting of the potentiometer, which, in turn, adjusted the programming to a desired value. The downside of such a scheme was that (i) it was too invasive and prone to infection and (ii) there was a risk of fluid ingress into the IMD via the opening, which resulted in short circuits and, thus, device malfunction [111]. These were the main reasons why modern IMDs ended up having a wireless-communication interface [53]. This integral IMD component, however, paved the way for allowing *any* entity to access the implant, including malicious entities.

1.1 IMD systems

Modern IMDs are autonomous, battery-powered devices with extremely high safety and reliability constraints. They are typically designed to operate for a long period of time (up to a decade or so) while implanted in the human body. These devices work in a closed-loop fashion by providing some form of stimulation based on monitoring one or more physiological signals. Even though IMDs come in various flavors,

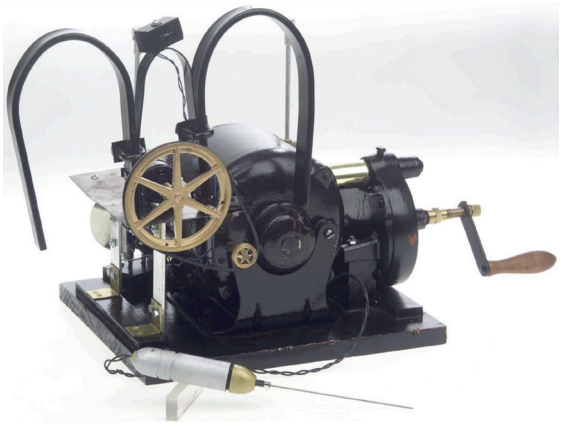


Figure 1.1: *Hyman's artificial pacemaker. © Vienna Museum of Science and Technology, Austria.*

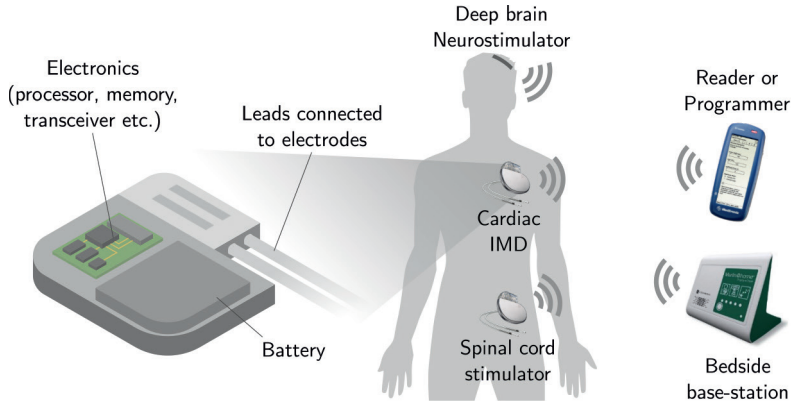


Figure 1.2: *Typical commercial IMDs*

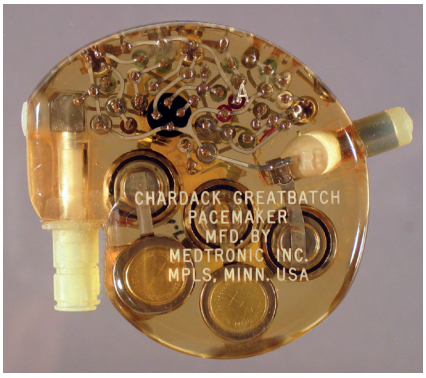


Figure 1.3: *One of the earliest IMDs: the Medtronic Chardack-Greatbatch pacemaker from the early 1960s. The opening on the right allows the programming of the device by using a Keith needle. © Collections of the Bakken Museum, Minneapolis, USA.*

i.e., for different target applications, they are very similar in their design. Typical components found in modern IMDs are shown in Figure 1.2. The sensors acquire the physiological data via the electrodes, which is processed by the IMD processor to determine if the patient requires electrical stimulation. The IMD also requires a memory unit to store treatment parameters, sensory information etc. Another important component is the battery to power the IMD.

As discussed above, to support and enhance the treatment capabilities of these devices, modern IMDs are equipped with wireless connectivity via a transceiver. Using this interface the IMD can communicate with an external reader/programmer[1] or a base station for e.g., local and/or remote monitoring of patient health, per-

[1] In this thesis, the term *reader* will be used for any device that is able to directly communicate with the implant.

forming a device test, reading sensory information, updating IMD settings and/or firmware, and so on [139]. The remote monitoring aspect allows better treatment of chronic conditions with significant cost reduction because of less frequent visits to the physician and less need for hospitalization. It also helps in the early detection of potential medical issues because of the possibility of quick access to the medical device [2]. However, these wireless capabilities – though greatly advantageous – make it possible for malicious entities to communicate with the device without the knowledge or cooperation of the victim. This vast expansion of the *attack surface* leads to a number of serious issues, e.g., private-data theft, misdiagnosis, physical harm etc.

1.2 Current IMD security problems

In 2016, MedSec Holdings and Muddy Waters Research disclosed their findings concerning cybersecurity vulnerabilities discovered in St. Jude Medical implantable cardiac devices [20]. A bitter litigation war soon ensued between the two sides, and in the aftermath, the stock price of the manufacturer plummeted by 10 percent [20]. Although security flaws of undisclosed IMDs have been reported in the past, this was the first time a security firm went public without disclosing the issues first to the IMD manufacturer giving them due notice. Their rationale for doing so was – they claimed – to wake up the manufacturer and force them into action [20]. In the following years, cybersecurity issues in medical devices were reported by the US government to have risen significantly [31–33]. However, modern IMDs still lack essential security provisions, a situation that – if left unchecked – threatens to make IMDs very hazardous devices in the years to come.

In a world becoming rapidly conscious of cybersecurity attacks and the need for data privacy, which has led to serious steps like the EU General Data Protection Regulation 2016/679 (GDPR), the slow reflexes of the IMD industry can be attributed to a number of reasons.

Firstly, this has been a niche industry historically having *no concerns* for or expertise on cybersecurity aspects. The earliest guidance from the FDA on securing wireless medical devices was issued as late as 2013 [49]. Thus, this is still a transition period for IMD manufacturers and is, to a point, reasonable.

Secondly, the *highly resource-constrained nature* of IMDs could be perceived as prohibitive for incorporating mainstream security provisions, as necessitated by the modern cybersecurity landscape, while maintaining a high device autonomy and a small form factor [9, 62].

Thirdly, the *steep (re)certification cost* of mission-critical and deeply embedded devices, such as IMDs are, is a major impediment to rehashing IMD design to include security provisions [44]. Though this is a valid concern, delaying incorporating IMD security provisions is a very short-sighted strategy in view of the prospective loss of

life and of ensuing market sales due to successfully mounted cybersecurity attacks in the future. The St. Jude Medical case serves as a cautionary tale of this fact.

Lastly, the medical-functionality codebase of IMDs, historically, has been slow to change, driven by a need for high reliability and by the sheer fact that little functional change is necessary in such deeply embedded devices. However, imbuing IMDs with adept security involves the introduction of a secondary, security codebase from scratch. We anticipate that this new codebase – besides the aforementioned feasibility challenges – more crucially will impose a *change of pace in IMD code updates* but also in code *maintainability*. We foresee more frequent updates to cope (a) with the virtually unsecured IMD designs, and (b) with the rapidly expanding attack surface of IMDs since they are now wirelessly accessible via end-user smartphones, tablets and – indirectly – the Internet [158]. IMD security, thus, requires a design-paradigm shift and also is suspect to introducing new, perpetual costs for IMD-code maintenance, which should be covered by the IMD industry.

1.3 Thesis scope and contributions

The primary goal of this thesis is to improve the current state of affairs in the IMD-security domain, with the focus on **secure reader-IMD communication**, which has a direct impact on patient *safety* and *privacy*, and **assessing the economic viability of adding security to IMDs**. Overall, this thesis makes the following contributions:

- A systematization of knowledge of the so-called Zero-Power Defense (ZPD) mechanisms, which protect against battery-depletion (or battery Denial-of-Service) attacks. These attacks have disastrous consequences for the patient's wellbeing, while at the same time they are among the simplest to mount from an attacker's perspective. This thesis raises essential design considerations for employing ZPD techniques in commercial IMDs, offers a critical review of such techniques found in literature and, subsequently, gives crucial recommendations for developing comprehensive ZPD solutions.

- IMDfence, a security protocol for IMD ecosystems that provides a comprehensive yet practical security portfolio, which includes availability, non-repudiation, access control, entity authentication, remote monitoring and system scalability. The protocol also allows emergency access that results in the graceful degradation of offered services without compromising security and patient safety. The performance of the security protocol as well as its feasibility and impact on modern IMDs are extensively analyzed and evaluated.

- SecureEcho, a device-pairing scheme based on MHz-range ultrasound that establishes trust between the IMD and an external reader. In addition, SecureEcho protects against battery-depletion attacks without requiring any energy

harvesting, which significantly reduces the IMD design complexity. We also provide a proof-of-concept implementation and a first ever security evaluation of the ultrasound channel, which proves that it is infeasible for the attacker to eavesdrop or insert messages even from a range of a few millimeters.

- An assessment of the economic repercussions of securing IMDs by employing the concept of technical debt (TD) on the evolving IMD software. This is extremely relevant since IMD manufacturers have only very recently started taking cybersecurity threats seriously, a move that will force development teams to overhaul IMD designs and grow sharper reflexes in an industry that has historically opted for small, careful steps. Thus, valid concerns arise regarding the technical feasibility but, chiefly, the economic viability of adding security to IMDs.

1.4 Thesis organization

This thesis is structured as follows: In Chapter 2, we provide an overview of the IMD threat landscape and highlight the critical security concerns and attacks [146]. This is achieved by performing a threat-modeling analysis based on *attack trees*, which provides a comprehensive and highly structured picture of the strengths and weaknesses of the IMD systems.

In Chapter 3, we extensively review works from literature that focus on protecting IMDs against battery-depletion attacks [147, 149]. We analyze these works and highlight their shortcomings after formulating design considerations. Furthermore, we provide recommendations towards implementing practical ZPD implementations. These include, among others, the concept of adaptive ZPD, which facilitates bedside-base-station operation, and the standalone ZPD module, which reduces the IMD-certification effort and, consequently, the time to market.

We then propose a novel security protocol for IMD ecosystems, IMDfence [144], in Chapter 4. We demonstrate that our approach offers a meticulous coverage of security requirements that are critical to these systems. These include, among others, access control, non-repudiation, user authentication, bedside-reader operation and system scalability. We show that IMDfence does not introduce any noticeable overheads in the implant, and it has the ability to support ZPD against battery-depletion attacks. We also propose an offline version of IMDfence, which employs an out-of-band (OOB) channel to enable IMD access in the absence of Internet or during emergencies.

In Chapter 5, we dive into the details of the OOB-pairing approach introduced in Chapter 4. We present SecureEcho, an ultrasound-based secure device-pairing scheme for reader-IMD systems that inherently provides protection against battery-depletion attacks [145, 150]. We show that the ultrasound channel used in the pairing process is sufficiently secure at MHz-range frequencies. We also demonstrate

a proof-of-concept implementation of the passive circuit that enables the pairing process and ZPD. Furthermore, we perform a detailed comparison of SecureEcho with the traditional RF-energy-harvesting-based ZPD schemes based on the design considerations formulated in Chapter 3.

In Chapter 6, we discuss a novel methodology to quantitatively analyze the cost of adding security in the existing IMDs from the perspective of embedded-software technical debt (TD) [151]. We discuss our experiment design in detail, which is based on a synthetically constructed IMD codebase. We then calculate TD-related metrics based on these software versions and make future predictions on software development costs and the economic viability of adding security.

Finally, in Chapter 7, we summarize the findings and the scientific contributions of this thesis, and highlight the potential directions for future research.

CHAPTER 2

"Security is not a product – it's a process. Attack trees form the basis of understanding that process."

Bruce Schneier, Dr. Dobb's Journal, December 1999

IMD threat landscape

M. A. Siddiqi, R. M. Seepers, M. Hamad, V. Prevelakis, and C. Strydis, "Attack-tree-based Treat Modeling of Medical Implants," in *PROOFS 2018. 7th International Workshop on Security Proofs for Embedded Systems,* ser. Kalpa Publications in Computing, vol. 7. EasyChair, 2018, pp. 32–49.

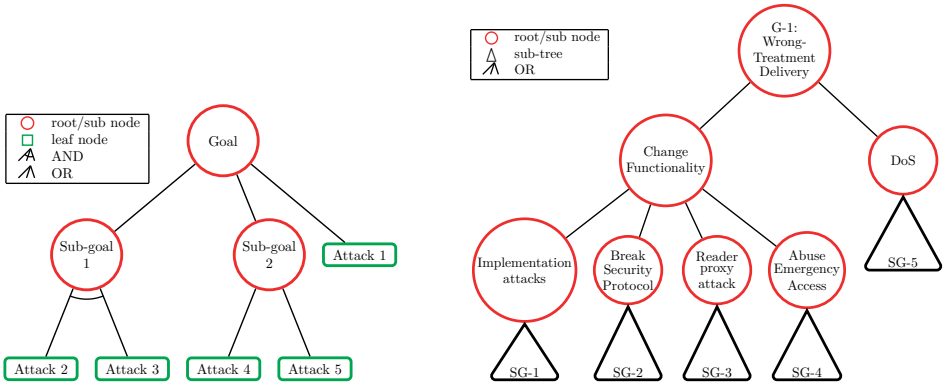
In this chapter, we provide an overview of the IMD threat landscape and highlight the critical security concerns and attacks. This is achieved by using attack-tree-based threat modeling, which is a non-exhaustive but structured approach that can be employed for finding vulnerabilities in this very ad hoc field. Threat modeling is a systematic process for identifying and categorizing threats and security vulnerabilities of a system from the adversary's perspective. It can be used to measure and improve the security of the system against current and future threats. Furthermore, it can be used to identify the adversary profile, the valuable assets of the system, the points of potential weakness and the most applicable threats. This approach can help us gain a better insight into the mindset and goals of the attackers and where security experts should spend effort considering the type of attacks expected from the attacker profile [137]. Threat modeling in a tree structure presents a comprehensive overview of the vulnerabilities and it can be used to analyze the different attack pathways in a structured way.

In this chapter, we establish (i) attack trees for the very particular case of IMDs, which, to the best of our knowledge, is the first work formulated for these devices. The intention is to create a constantly expanded reference point by and for the whole IMD community. Furthermore, we assess (ii) these trees by evaluating the security of three recent secure IMD-communication protocols from literature. We subsequently give recommendations on how to improve the security of these sample protocols.

The rest of the chapter is organized as follows. A brief background on attack trees is provided in Section 2.1. In Section 2.2, detailed, IMD-specific attack trees are constructed after defining the system and attacker models. Background information on the protocols chosen for our threat-analysis approach is provided in Section 2.3. In Section 2.4, the example protocols from Section 2.3 are evaluated using the above threat-analysis approach. Recommendations are given based on our findings in Section 2.5 and the related work is highlighted in Section 2.6. We conclude the discussion in Section 2.7.

2.1 Background on attack trees

Attack trees were proposed by Schneier [137] as a method to describe the security of any system. These constructs aid in improving the security or evaluating the impact of new attacks on security. Attack-tree-based analysis helps to determine the vulnerability of a system against any specific type of attack and can rank different types of attacks based on their likelihood. It is also useful in enumerating the security assumptions of the system. In the case of a system modification e.g., generic improvements or implementation of countermeasures against a threat, attack trees help in evaluating the resulting impact on security. Since smaller attack trees can fuse in larger trees, the resulting scalability helps in determining the coverage and efficiency of any countermeasure. Moreover, they can help in efficiently allocating the security budget available and can also shed light on the resources, level of access



***(a)** Example Attack tree* ***(b)** G-1: Wrong-treatment delivery*

***Figure 2.1:** Attack trees for IMDs*

and skills required by the attacker to perform certain attacks. Lastly, compared to other threat-modeling methods (such as STRIDE, PASTA, CVSS, etc.), attack trees are relatively simple to use if one has a thorough understanding of the system [141]. Based on the above features, we select attack trees from a multitude of available options to perform an initial threat-landscape assessment of a very niche field.

These constructs illustrate the attacks in a tree structure as shown in Figure 2.1a. Each tree contains a *root node*, which represents the final goal of the attack, *intermediate nodes* (sub-goals), which define different stages of the attack that lead to the root, and *leaf nodes*, which represent atomic attacks. Boolean gates are used to explain whether a node in a tree requires achieving *all* of its sub-nodes (*AND* node), or *any* of its sub-nodes (*OR* node). An attack scenario will contain a minimum set of leaves that leads to a successful traversal to a root.

Given the largely unsecured or – worse yet – ad hoc manner in which security is being added to modern IMDs, attack trees offer a more structured method for designing and evaluating IMD security. This approach does not guarantee completeness but takes a methodical and scholastic approach towards IMD security, which is hoped to result in uncovering more blind spots.

2.2 IMD threat modeling

We first define our system and attacker models, after which we follow the methodology from [137] to construct IMD attack trees. The first step is to identify the attacker goals. Each goal will result in a separate attack tree. We then identify possible attacks pertaining to each goal to populate the tree. Existing trees can be reused as sub-goals to form part of a bigger tree.

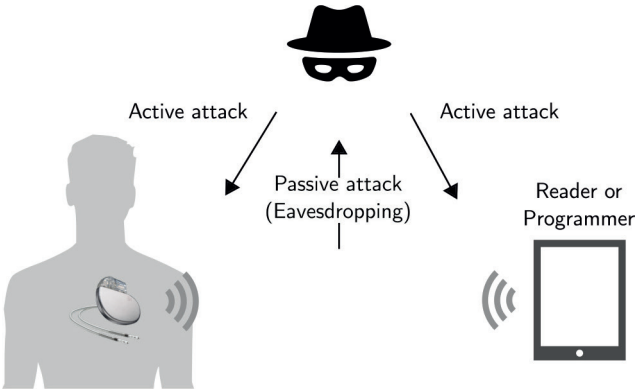


Figure 2.2: *Attacker model*

2.2.1 IMD system model

We consider an IMD I capable of wireless communication with an external reader R. Both entities fall within the boundary of our system model. We assume that there is only wireless (non-physical) access to I while implanted in the body, whereas physical access to R is possible. Without loss of generality, we assume that the implant application supports more than one user role in terms of lowest to highest permission levels. For example, a nurse may only be allowed to read data related to the operation of the implant, whereas a treating physician may also be allowed to suspend or resume the implant functionality or modify it for therapy updates.

2.2.2 Attacker model

In order to evaluate IMD security, we assume an attacker A whose aim is to prevent patient treatment, perform data manipulation, or steal private patient data. This is further elaborated in Section 2.2.4. Necessarily, A can either be an *outsider* or from various insiders with different security privileges (e.g., nurses, physicians, technicians etc.) [132]. Furthermore, we assume that A is active, i.e., has full control of the channel. Thus, A can eavesdrop, modify, block or replay messages between R and I, in addition to forging new ones (see Figure 2.2). We note that these are rather conservative assumptions given the current state of the art in IMD communications. However, we consider worst-case conditions to cover any future changes in this aspect.

2.2.3 Security services and features

In light of the above, the IMD communication protocol must satisfy the fundamental security services of CIANA: *Confidentiality*, *Integrity*, *Availability*, *Non-repudiation*[1] and *Authentication*. In addition, the protocol must provide the following features of particular importance to IMDs: *Access Control*, *Key Management*[2], *Key Freshness* and *Perfect Forward Secrecy (PFS)*. Also, *Emergency Access* is a crucial feature specific to IMDs, which allows paramedics access to the IMD during emergencies without compromising security. Strictly speaking, these features fall under the CIANA model, but are explicitly mentioned here as they are of special importance for IMDs, e.g., emergency access can be considered as falling under availability. These services will be revisited in detail in Chapter 4.

2.2.4 Attacker goals

The attacker goals subject to the above System and Attacker models are categorized below.

2.2.4.1 Modification of IMD operation / Wrong-treatment delivery

Many modern IMDs are working in a closed-loop fashion, effecting some form of intervention (e.g., electrical stimulation) to a particular health issue (e.g., heart arrhythmia, epileptic seizures, chronic pain, tremor etc.). Common examples are modern-day implantable pacemakers. Modification of this functionality may result in the prevention of stimulation for treatment purposes. It may also result in over-stimulation, which could cause tissue damage. Attacker A can cause these modifications by forcing the processor to execute a different/modified binary leading to incorrect IMD functionality, or by making it run an infinite loop resulting in thermal hot-spots and battery drain. Alternatively, A can cause the sensors to read incorrect physiological data, subsequently resulting in incorrect calculations by the processor. A can also try to modify the clock frequency of the system resulting in incorrect duration of treatment and untimely triggering of stimulation.

A can also force critical IMD resources to remain unavailable when treatment action is required (e.g., during cardiac arrest) through various Denial-of-Service (DoS) attacks. A can, for instance, force the processor to run an infinite loop at full frequency resulting ultimately in energy loss and IMD shutdown (*Battery DoS*). The IMD may also be repeatedly requested to establish a secure wireless channel using incorrect credentials. This will cause repeated execution of the same authentication protocol for analyzing the request, which will – in turn – result in battery drain.

[1] Non-repudiation is a valid concern since we assume the possibility of insider attacks, malpractices etc. For instance, a method is needed to perform computer forensics in case a patient dies or has medical issues due to a mis-configured pacemaker by a careless physician, and so on.

[2] In addition to generating keys, here key management also includes the revocation, replacement and addition of new readers.

Moreover, repeated communication requests may prevent the IMD from performing its life-critical, primary task (*Function DoS*). A can also block the reader/IMD communication channel by constantly sending valid or invalid messages resulting in *Jam DoS*[3] [139].

2.2.4.2 Data forging

Another goal of A could be to *forge* sensitive patient data, e.g., data-logs stored in I, which can indirectly lead to patient/doctor misinformation, incorrect diagnosis and subsequent incorrect treatment. Data forging could also be the goal of an insider to cover up medical mistakes (e.g., wrong diagnosis). A can perform this by modifying the IMD memory to store incorrect data, or by manipulating the communication packets exchanged between R and I.

2.2.4.3 Data theft

The aim of A can also be to *steal* private patient data, which can indirectly lead to problems such as social segregation, extortion, blackmail and more [139]. This can happen if A steals data from the IMD memory, or if he/she eavesdrops on such data exchanged between R and I.

2.2.5 Attack trees for IMDs

For each threat identified in Section 2.2.4, we now present the attack trees per attacker goal G-x.

2.2.5.1 G-1: Wrong-treatment delivery

The high-level tree to reach this goal is shown in Figure 2.1b. We use Figure 2.3 to expand sizable sub-goals (SG-x). To also highlight the benefit of our approach, we coarsely assign the likelihood of attack (*L=Low, M=Moderate, H=High*) to all the leaf-nodes and propagate the resulting effect up towards the root node. The likelihood can be a function of cost, type of equipment required etc. [137]. Wrong treatment could be achieved either by changing the IMD functionality or through DoS. The IMD functionality can be modified by using implementation attacks, breaking into the secure channel (to forge communication packets), using a Reader proxy or by abusing the emergency-access mode.

SG-1: Implementation attack[4]: This sub-tree consists of side-channel attacks, fault injection and attacks that exploit implementation flaws. A can employ side-channel attacks to e.g., recover the factory-installed key from a stolen R [93]. This

[3] The chance of a jam DoS harming IMD operation is extremely low since the communication between R and I pertaining to critical treatment updates is very infrequent and is usually held in a controlled environment [164].

[4] SG-1.1 and SG-1.2 are more likely to be accomplished against R than I. The likelihood annotations of these sub-goals, however, pertain to I for consistency.

SG-1 Implementation attack
OR
SG-1.1 Side-channel attack [93] (L)
OR
SG-1.1.1 Power analysis [114] (L)
OR
SG-1.1.1.1 Simple Power Analysis (L)
SG-1.1.1.2 Differential Power Analysis (L)
SG-1.1.1.3 Template Attack (L)
SG-1.1.1.4-N ...
SG-1.1.2 Timing analysis [114] (L)
SG-1.1.3-N ...
SG-1.2 Fault injection [12] (L)
OR
SG-1.2.1 Change sensor/actuator functionality through EMI [79] (L)
SG-1.2.2 Clock glitch attack (L)
SG-1.2.3 Power glitch attack (L)
SG-1.2.4-N ...
SG-1.3 Identify Implementation flaws (M)
OR
SG-1.3.1 Protocol-implementation flaw. See SG-2. (H)
SG-1.3.2 Buffer overflows (M)
SG-1.3.3 Race condition between different processing cores (L)
SG-1.3.4 Bug or flaw in application-code compiler. For deliberate errors see SG-1.4. (L)
SG-1.3.5 Flaws in data sanitization (M)
SG-1.3.6-N ...
SG-1.4 Insider Attack (L)
OR
SG-1.4.1 Attack Tool Chain
OR
SG-1.4.1.1 Compiler (L)
SG-1.4.1.2 Libraries (L)
SG-1.4.1.3 Run-time environment (L)
SG-1.4.1.4-N ...
SG-1.4.2 Attack application (L)
SG-1.4.2.1 Exploit improper access control [65] (L)
SG-1.4.2.2-N ...

SG-2 Break security protocol
OR
SG-2.1 Identify flaw in Security Protocol (H)
OR
SG-2.1.1 Identify flaw in encryption alg. (M)
OR
SG-2.1.1.1 Identify flaw in cipher (L)
SG-2.1.1.2 Identify flaw in RNG(s) to mount a replay attack (L)
SG-2.1.1.3 Identify key re-use (M)
SG-2.1.1.4 Identify lack of randomness (M)
SG-2.1.1.4.1 In biometrics [93] (M)
SG-2.1.1.4.2 Due to ECC usage [93] (M)
SG-2.1.1.5-N ...
SG-2.1.2 Identify flaw in handshake (H)
OR
SG-2.1.2.1 Man-in-the-middle attack [93] (M)
SG-2.1.2.2 Reflection attack [93] (H)
SG-2.1.2.3 Key confirmation attack [114] (L)
SG-2.1.2.4 Replay attack [62, 94, 114] (H)
SG-2.1.2.5-N ...
SG-2.2 Obtain legitimate master key for IMD (H)
OR
SG-2.2.1 Brute-force master key used in cipher (L)
SG-2.2.2 Acquire key through social engineering (H)
SG-2.2.3 Steal key (from a used IMD) using side-channel attack. See SG-1. (L)
SG-2.2.4 Insider attack. See SG-1.4 and G-2.3. (L)
SG-2.2.5-N ...

SG-3 Reader Proxy Attack
OR
SG-3.1 Hack into patient laptop/smartphone (L)
OR
SG-3.1.1 Perform remote physiological-signal measurement using camera etc. [179] (L)
SG-3.1.2 Start IMD-control application (if available) (L)
SG-3.1.3-N ...
SG-3.2 Use legitimate medical equipment to aid in remote attack [94] (M)
SG-3.3-N ...

SG-4 Abuse Emergency Access
OR
SG-4.1 Wait until emergency mode is triggered (L)
SG-4.2 Evoke emergency directly by creating a stressful incident (M)
SG-4.3 Toggle device to Emergency mode (M)
OR
SG-4.3.1 Exploit proxy device (M)
OR
SG-4.3.1.1 Physically remove the proxy device (M)
SG-4.3.1.2 Remotely attack proxy device. See SG-2. (M)
SG-4.3.2 Exploit token-based access (M)
OR
SG-4.3.2.1 Gain physical access to the token (M)
SG-4.3.2.2 Brute-force emergency password (L)
SG-4.3.3 Exploit distance-bounding protocol (M)
OR
SG-4.3.3.1 Body-coupled channel: Capture on-body (electric) signals (L)
SG-4.3.3.2 Vibration-based: Cause vibrations on the body (L)
SG-4.3.3.3 Magnetic switch (M)
OR
SG-4.3.3.3.1 Toggle magnet when proximal to I (M)
SG-4.3.3.3.2 Use strong magnet for remote attack (L)
SG-4.3.4 Biometrics (L)
OR
SG-4.3.4.1 Obtain biometrics from subject (L)
OR
SG-4.3.4.1.1 Remote measurements. See SG-3. (L)
SG-4.3.4.1.2 Physical measurements (touching patient) (L)
SG-4.3.4.2 Brute-force the biometric (L)
SG-4.3.4.3 Implementation error in (biometric) cipher/authenticator (L)
SG-4.3.5 Criticality-awareness-based access (M)

SG-5: DoS Attack
OR
SG-5.1 Disrupt channel (H)
OR
SG-5.1.1 Signal jamming on IMD-communication frequency (H)
SG-5.1.2 Keep the comm. channel busy with requests
OR
SG-5.1.2.1 Repeatedly request connection to I using normal-mode protocol (even if authentication fails) (H)
SG-5.1.2.2 Repeatedly request to activate emergency mode (even if this fails) (H)
SG-5.1.3 Send bogus packet or modify/drop packets to reset a session between R and I (H)
SG-5.2: Cause device malfunction (H)
OR
SG-5.2.1 Drain IMD battery [62] (H)
OR
SG-5.2.1.1 Overloading connection requests (H)
SG-5.2.1.2 Execute additional code on IMD. See SG-2, SG-3 and SG-4. (L)
SG-5.2.1.3-N ...
SG-5.2.2 Overheat device (L)
OR
SG-5.2.2.1 Execute additional code on IMD. See SG-5.2.1.2. (L)
SG-5.2.2.2 Provide continuous stimulation. See SG-2, SG-3 and SG-4. (L)
SG-5.2.2.3-N ...
SG-5.2.3 Prevent I to execute its main application. See SG-5.2.1.1. (H)
SG-5.2.4 Replay previously captured commands from R to *turn off* the therapy [62] (H)
SG-5.3: Insider attack. See SG-1.4. (L)

Figure 2.3: *Textual representation of IMD Attack trees for sub-goals of G-1*

can be done by e.g., measuring the power consumption of a processing core when running the crypto algorithm, observing the run-time behavior of an algorithm implementation etc. [114]. In the case of fault-injection attacks, A can affect the IMD operation by e.g., changing the sensor and actuator functionality through electromagnetic interference (EMI) [79]. A can also inject fault in the IMD clock source, e.g., crystal oscillator, to induce additional toggles within the clock period (clock glitch attack). This may result in timing failure of certain portions of the IMD. Barenghi et al. have listed various fault-injection attacks, which can be utilized against Reader/IMD systems [12]. A can also attempt to exploit implementation flaws in the IMD. These flaws could lie in the security protocol, e.g., in nonce[5] generation, which create opportunities for replay attacks (see SG-2), or in the device firmware causing buffer-overflow exploits. Race conditions between different computing components of the IMD and errors in the application-code compilers also open up opportunities for exploitation. Flaws in data sanitization allow an *illegal* value to destabilize the system, e.g., if there is an option to select an encryption algorithm among various choices, then inputting a negative crypto-algorithm identifier may cause unauthorized code execution. Implementation attacks can also be a viable option for a trusted entity that is malicious, e.g., manufacturer, developer, and so on. The insider, in this case a developer, can attack the tool chain for the software used in R and/or I by modifying the application compiler, libraries or the run-time environment, or, the application itself.

SG-2: Break security protocol: Another option for A could be to break the security protocol between R and I. A can, for instance, identify flaws in the used cipher for data confidentiality or the used random-number generator (RNG) for nonce generation (for replay attacks). A can also look for cases of key re-use, lack of randomness in biometrics or lack of randomness due to the use of Error-Correcting Codes (ECCs) if the protocol employs fuzzy cryptographic primitives [93]. Alternatively, A can find flaws in the protocol handshake process. Marin et al. [93] proposed attacks specific to physiological-signal-based security protocols. Such protocols rely on the reader-IMD pair to measure a biometric/physiological signal from the patient's body. Access is allowed based on the similarity of these measurements. They showed that a well-cited Dynamic-Cardiac-Biometrics (DCB)-based protocol (H2H) [131] had weaknesses against *Man-in-the-middle (MITM)* and *reflection attacks*. A can also opt for other common attacks such as *replay attacks* and/or *key-confirmation attacks* [114]. Halperin et al. [62] demonstrated successful replay attacks on a commercial implantable cardioverter defibrillator (ICD) over a short-range communication channel by replaying previous messages sent by the reader. Marin et al. [94] showed that adversaries can launch successful replay attacks on multiple commercial IMDs over both short- and long-range communication channels.

[5] A freshly generated random number that is used only once.

Besides, A can try to obtain the legitimate IMD cipher master key in order to break the security protocol. A can, for instance, attempt a brute-force attack or use social engineering, e.g., by employing blackmail or phishing on a trusted entity or the IMD manufacturer.

SG-3: Reader proxy attack: This sub-goal pertains to the scenario where A uses legitimate equipment in place of the reader. For protocols based on physiological signals, e.g., heartbeats (see SG-2 above), A can hack into the patient smartphone/laptop camera and measure subtle color variations of the patient skin to detect heartbeats using remote photo-plethysmography (rPPG) [139, 179]. A can also hack the smartphone to run an IMD control application, if available (see the control application in [162], for example). Another approach could be to buy an inexpensive, compatible base station that only gathers telemetry data from the IMD and sends it to the hospital to facilitate remote monitoring. A can use it to activate the IMD and then use his/her own equipment to send malicious messages, as shown in [94] for a commercial IMD.

SG-4: Abuse emergency access: In the case of emergencies, the IMD should permit access to a paramedic's reader for immediate treatment despite the fact that they are likely unknown to each other and therefore do not share a secret key [138]. Emergency-access schemes found in literature mostly rely on a *touch-to-access* policy, which ensures that only the entities that can physically touch the patient for a prolonged period of time are allowed access to the implant [131]. In other words, it is infeasible for an attacker to get in close proximity to the patient, and even if that is the case, the patient can detect this and reject physical contact. Also, the attacker would then have far easier methods to harm the patient than via accessing the implant, e.g., by physically attacking the patient. Emergency-access schemes can be broadly categorized as follows [139]: **Proxy-based** schemes use an additional device in the possession of the patient, such as a smart phone, watch, etc. [121, 156]. The device is paired with the IMD and is used to authenticate the reader that is trying to communicate with the implant. In case of emergency, the device can be physically distanced from the patient in order to grant the reader unsecured access to the IMD. **Token-based** schemes rely on the patients having the IMD-access key or password with them, which is stored e.g., on a bracelet. During an emergency, a paramedic can access the IMD using this token. **Distance-based** schemes (e.g., [77, 124]) employ weak or out-of-band (OOB) signals for reader-IMD communication. These can either involve direct transfer of a session key, which would be hard for an attacker to eavesdrop, or they can require the devices to mutually prove proximity to one another. This can be done by using e.g., the human body as an OOB channel (using electric conductivity or vibrations), a magnetic switch (to disable security) etc. **Biometric-based** schemes (such as [131, 138]) rely on both the reader and IMD to measure a physiological signal from different parts of the patient's body (see SG-2 above). The devices are paired based on the similarity of these measurements.

Criticality-awareness-based schemes, unlike previous methods, do not follow a touch-to-access policy. In these schemes, I monitors patient vitals and triggers fail-open access in case of emergency.

Following this sub-goal, A can wait for the emergency to happen, evoke a fake emergency situation directly by creating a stressful incident, or toggle the device to this mode based on the type of access scheme employed by the IMD: A can attack the proxy device in case of a proxy-based scheme. Exploitation of token-based schemes would require an attacker to get access to a token that has the emergency password for the paramedics. For the devices relying on the human-body channel, depending upon the implementation, A can try to remotely capture on-body electric signals, cause vibrations on the body by calling the patient cell-phone etc. For the devices employing a magnetic switch, A can pass a magnet over I to disable security when in close proximity. A relatively expensive alternative is to use a strong magnet in case of a remote attack. Regarding the biometric-based schemes, A can opt for performing measurements remotely (as discussed in SG-3) or physically while touching the patient e.g., by impersonating a nurse. A can also opt to brute-force the biometric if it lacks perfect entropy or try to find implementation flaws in the security primitives employed in biometric-based encryption schemes, e.g., fuzzy vault etc. When it comes to Criticality-awareness-based schemes, A can try to fool I to believe that it is in a medical emergency.

SG-5: Denial of service: To achieve DoS, A can disrupt the wireless channel to block communication between R and I in order to prevent medical intervention. This can be done by jamming the IMD frequency band, by repeatedly requesting connection to I even if the authentication fails, or by sending/modifying/dropping packets to reset a communication session between R and I. A can also cause malfunction or shorten lifetime of I by draining battery [62] by overloading I with connection requests or by running additional code via code injection (using SG-2, SG-3 and SG-4). Marin et al. [94] carried out battery DoS for certain pacemakers by switching the devices from standby to (energy-consuming) interrogation mode with relative ease. Through continuous execution, the additional malicious code can also overheat the device, e.g., by causing continuous stimulation. The overloading of connection requests from A can also result in the inability of I to execute its main application.

2.2.5.2 G-2: Data forging

The attack tree for data forging (G-2) is shown in Figure 2.4. Note that the listed attacks are common to those discussed in Section 2.2.5.1. In addition, an insider e.g., a doctor or nurse can try to attack the logistics, e.g., by malicious handling of the treatment logs, in order to cover up medical mistakes.

G-2: Data Forging
OR
G-2.1 Modify I memory data after initiating communication with it. See SG-2, SG-3 and SG-4. (L)
G-2.2 Inject/modify communication packets between R and I [62]. See SG-2, SG-3 and SG-4. (L)
G-2.3 Insider attack (by attacking operations/logistics) (L)
OR
G-2.3.1 Forging/mishandling of treatment logs (L)
G-2.3.2-N ...
G-2.4-N ...

G-3: Data theft
OR
G-3.1 Steal private data (data logs etc.) (H)
OR
G-3.1.1 Compromise the reader where logs are downloaded (M)
G-3.1.2 Steal data by tampering with a used implant (L)
G-3.1.3 Retrieve private data from the IMD memory after initiating communication with the implant. See SG-2, SG-3 and SG-4. (L)
G-3.1.4 Eavesdrop on communication between R and I [62]. See SG-2, SG-3 and SG-4. (H)
G-3.1.5-N ...
G-3.2 Investigate which IMD type is implanted inside the patient by finding the IMD identifier (H)
OR
G-3.2.1 Retrieve identifier by initiating communication with the implant. See SG-2, SG-3 and SG-4. (H)
G-3.2.2 Eavesdrop on communication between R and I. See SG-2, SG-3 and SG-4. (H)
G-3.2.3 Retrieve identifier/model number etc., by looking at the reader (H)
G-3.2.4-N ...
G-3.3 Insider attack. See G-2.3. (L)
G-3.4-N ...

Figure 2.4: *Textual representation of IMD Attack trees for G-2 and G-3*

2.2.5.3 G-3: Data theft

When it comes to data theft (see Figure 2.4), A could either be interested in stealing treatment-related data/logs etc., or just the nature of the medical condition itself. Stealing of private data can be done by compromising the reader, by stealing a used implant or by using attacks from Section 2.2.5.1 to hack into I and modify the IMD memory or, finally, to eavesdrop and decrypt communications between R and I [62]. In order to discern the nature of the medical condition, A can try to investigate the type of IMD implanted in the patient by finding the IMD identifier. This can be done by looking at the model/type of R. Alternatively, A can opt for remote attacks using SG-2, SG-3 and SG-4.

2.3 Example IMD protocols

In order to demonstrate our proposed threat-modeling approach, we use the protocols proposed in [118, 138, 164], which were designed to enable secure communication between R and I. These protocols were selected for our analysis because they are custom-made for low-power IMD systems and entail state-of-the-art research concepts such as zero-power defense [62], dynamic-biometrics-based security, touch-to-access policy, emergency access, and so on. Moreover, it is well known that the IMD manufacturers rely on "security through obscurity" by concealing the protocol specifications [94], which is another reason to evaluate the above protocols from academia. The three protocols are summarized below and are denoted by P-Sec, P-KeyEx and P-Auth, respectively, for brevity.

2.3.1 P-Sec: Lightweight secure communication protocol

The main purpose of this protocol is to ensure confidentiality, integrity and mutual authentication of the messages exchanged between R and I. It uses a lightweight symmetric block cipher for data confidentiality. Moreover, cipher-based Message

Table 2.1: *Table of Notations*

Notation	Definition
ID_A	Identifier of entity A
N_A	Nonce generated by A
K_{pubA}/K_{prA}	Public/private key pair of A
K_{AB}	Pre-shared symmetric key between A and B
K'_{AB}	Short-term symmetric key between A and B
K_A	A secret only known to A
P_A	Privilege information of user/card A
k	Difficulty of solving a client puzzle
$x < i >$	i^{th} bit of a bitstring x
$x < i : j >$	Bit sequence $x < i >, ..., x < j >$
t	Time stamp
T	Lifetime of reader-card authentication
$\{\}_{K_{AB}}$	Authenticated encryption* using K_{AB}
$\mathsf{h}()$	hash function
$\mathsf{kdf}()$	key-derivation function
$\mathsf{MAC}_{K_{AB}}()$	MAC operation using key K_{AB}
$\mathsf{sig}_{K_{prA}}()$	Signature (of a hashed message) using K_{prA}
$\{\{x\}\}_w$	Fuzzy commitment of x using witness w (i.e., $ECC(x) \oplus w$)
$Cert_A$	Certificate consisting of ID_A, P_A, K_{pubA} and $\mathsf{sig}_{K_{prCA}}(ID_A, P_A, K_{pubA})$

* Such as Encrypt-then-MAC (EtM). Depending on the authenticated-encryption implementation, separate keys may be required for the encryption and MAC operations to prevent certain attacks and to ease key management [108]. However, these keys are not differentiated here for simplicity.

Authentication Code (MAC) is used for integrity and authentication. The protocol uses nonces in the MAC calculation to prevent replay attacks.

The notations used in this chapter are listed in Table 2.1. The notation $\{\cdot\}_K$ denotes *authenticated encryption* using a key K, which, in addition to confidentiality, also provides message authentication and data integrity by computing a MAC. The protocol steps are shown in Figure 2.5. R initiates the protocol by sending ID_R, which is used by I to choose the correct key K_{RI}. I responds by sending N_I to R. R then generates N_R and encrypts the command (CMD) and the nonces using K_{RI}. R then sends N_R together with the encrypted command to I. I checks if the message has been received correctly and was indeed sent by R by locally calculating the MAC and checking its equality to the received value. I aborts the protocol in case the validation fails. Otherwise, it confirms R as a legitimate entity. I then decrypts and executes the CMD and sends the subsequent answer (ANS) in a similar fashion to how it was done for CMD in the previous step. R receives the message and calculates the local version of the MAC. R and I are considered mutually authenticated if both the MAC values are equal and as a result, the ANS is decrypted and processed by R. Otherwise, R drops the reply.

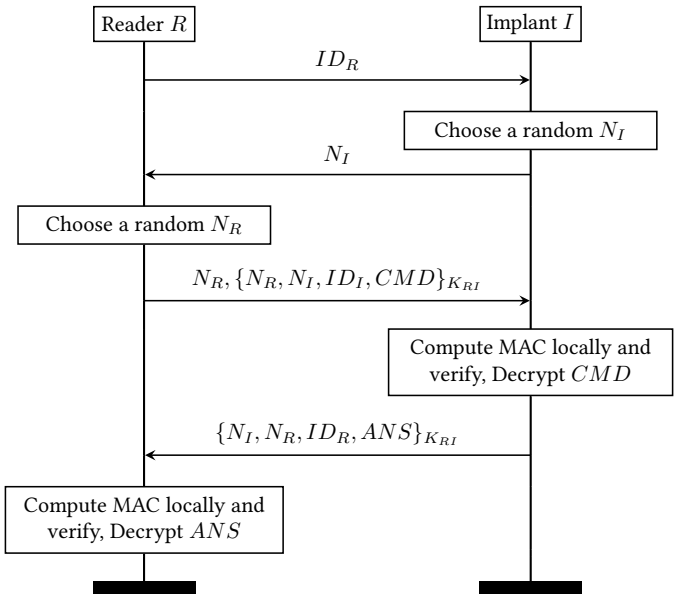


Figure 2.5: *P-Sec [139, 164]*

2.3.2 P-KeyEx: Lightweight authenticated key-exchange protocol

The purpose of P-KeyEx is to establish trust between R and I, and to perform key exchange for any symmetric-key-based data confidentiality protocol. It achieves this by using the cardiac Inter-Pulse Interval (IPI) [131], which is the time difference between two consecutive heartbeats, as an RNG. An IPI value is obtained by both R and I by simultaneously measuring a cardiac signal from the same person. Each of these values are used to derive time- and person-specific random numbers, which are referred to as *witnesses* w_R and w_I, respectively. This makes them useful in entity authentication and key exchange [138]. However, if I transports K_{RI} by encrypting it using w_I as symmetric key, R cannot decrypt it, since in practice w_R is not exactly equal to w_I. The protocol addresses this by employing a *fuzzy-commitment* scheme [74], which applies ECCs on K_{RI} before its encryption to counter the difference between w_R and w_I. The commitment operation $\{\{x\}\}_w$ of x using witness w is defined as $\{\{x\}\}_w = ECC(x) \oplus w$.

The protocol is depicted in Figure 2.6. Both R and I exchange ID_R and ID_I in order to bind K_{RI} to these identifiers upon successful exchange. To generate witnesses, R and I simultaneously obtain a block of IPIs and communicate the occurrence of any *heartbeat misdetection* using misdetection flags (m_R and m_I). In the case of a misdetection they replace the block of IPIs with fresh IPIs. This process is repeated until the gathered IPIs are enough to generate w_R and w_I. I generates a random K_{RI} (through its internal RNG) and fuzzy commits it using w_I, calculates

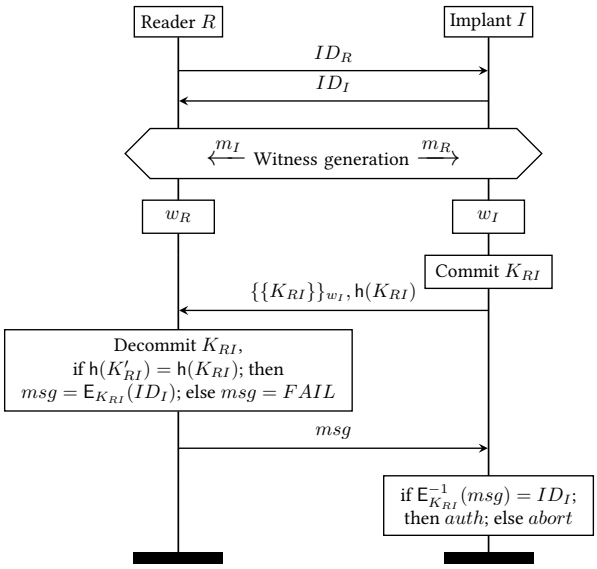


Figure 2.6: *P-KeyEx [138]*

a cryptographic hash of K_{RI} ($\mathsf{h}(K_{RI})$) for *data integrity*, and sends the entire message to R. R after receiving this message applies the inverse process of commitment and obtains K'_{RI} (where $K'_{RI} = K_{RI}$ iff $w_R \approx w_I$). R validates the correct transfer of K_{RI} by locally computing the hash $\mathsf{h}(K'_{RI})$ and comparing it to $\mathsf{h}(K_{RI})$ received from I. In case of a match, R encrypts ID_I with K_{RI} using its regular cipher and sends it to I. I decrypts the ID_I with K_{RI} using the same cipher. If the decrypted ID_I is correct then the key exchange is a success. At this point, both R and I are mutually authenticated since both have implicitly verified that $w_R \approx w_I$. If either of R's hash comparison or I's identifier comparison fails, then the protocol fails.

2.3.3 P-Auth: Biometric authentication protocol

Similar to P-KeyEx, the purpose of P-Auth is to authenticate the two entities using IPIs. However, as opposed to P-KeyEx, P-Auth does not perform symmetric-key exchange. The protocol steps are shown in Figure 2.7. P-Auth starts with a session key establishment using any suitable key exchange protocol. Both nodes measure the IPIs and generate random nonces N_R and N_I. These nonces are then exchanged after which both nodes calculate a hash of the locally measured IPIs and the received nonce. They exchange these hashes (H_R and H_I) *before* sharing their IPIs with each other. This prevents the peer node to replay the received IPI value. Both nodes then locally calculate the hash of the received IPIs and the locally generated nonce. If the resulting value is not equal to the received hash value, or the local and received IPIs are not similar enough, the authentication fails.

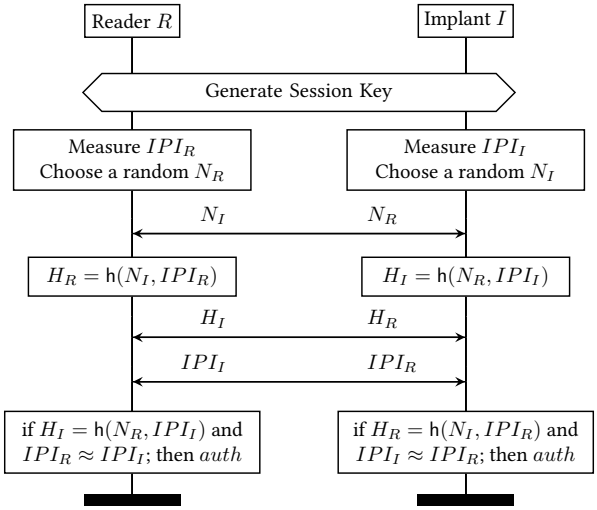


Figure 2.7: *P-Auth [118]*

2.4 Evaluation using attack trees

In this section, we evaluate the three protocols described in Section 2.3 by reflecting them against our attack trees introduced in Section 2.2.5. We will only use the portion of the attack trees relevant to the evaluation of security protocols. However, keep in mind that the trees are designed to have much broader applicability as these are generic trees for Reader/IMD systems.

2.4.1 P-Sec analysis

The goals/sub-goals relevant to the evaluation of P-Sec are SG-1, SG-2 and SG-5 (see Figure 2.3).

SG-1: Implementation Attack. These attacks are by definition implementation specific, yet, we discuss them here to highlight prominent ways in which the protocol implementations can be exploited, since some of the implementations are dependent on protocol architecture.

Looking at SG-1, we can instantly see that P-Sec's dependence on pre-installed keys[6] opens the door for A to perform side-channel attacks to steal the factory-installed keys from the reader **(SG-1.1)**. Obviously, any protocol implementation involving the storage of secret data is in principle susceptible to these attacks. Therefore, such pre-installed keys need to be properly protected. Protocol-implementation flaws **(SG-1.3.1)** e.g., incorrect nonce-generation implementation, also invite attacks from A. By construction P-Sec allows distinguishing individual users (or user

[6]It was not the objective of the authors of P-Sec to address key management, as explicitly mentioned in [164].

groups) because of pre-shared unique keys, thus facilitating access control. Hence an insider with valid authentication credentials would not be able escalate privileges if access control is implemented correctly **(SG-1.4.2.1)**.

SG-2: Break Security Protocol. We start the analysis of the protocol itself by first looking at the encryption algorithm using **SG-2.1.1**. The MISTY1 cipher employed by P-Sec was recently broken using *Integral Cryptanalysis* [172] **(SG-2.1.1.1)**, which makes the cipher-based RNG used for nonce generation vulnerable too **(SG-2.1.1.2)**, hence making the protocol susceptible to replay attacks. Moreover, for all block ciphers with a block size of 64 bits, including MISTY1, the likelihood of ciphertext block collisions increases if large amounts of data are encrypted with the same session key (in most modes of operation e.g., CBC, CFB, CTR etc.), thus making them vulnerable to *birthday attacks* [17]. Although these attacks are currently unrealistic for low data-traffic IMDs, they still warrant a fix considering that these devices are intended to remain implanted for many years. Therefore, the cipher implementation should make sure that it encrypts less data per session key and that the session keys are changed more frequently. Since, in P-Sec, MISTY1 is used in CTR mode for nonce generation using the same key for all the sessions, this makes it vulnerable to these attacks. One of the most important issues with the protocol is that the authentication key is the same as the session key (K_{RI}), which is re-used in every session **(SG-2.1.1.3)**. This means that we lose *PFS*, i.e., if this shared key is compromised in the future, then all the previous communication (which is recorded by the adversary) will also be compromised. In a device that is expected to operate for a long time, satisfying PFS has high significance. Moreover, the use of a single factory-installed K_{RI} makes it a *single point of failure*, which is undesirable [93].

We analyze the handshake using **SG-2.1.2**. Since the protocol relies on symmetric encryption, it is by design robust against MITM attacks **(SG-2.1.2.1)**. It should be noted, however, that because of symmetric encryption, the protocol does not guarantee *non-repudiation*. Moreover, the protocol was originally designed for a few readers with pre-distributed symmetric keys. There is no mechanism to introduce new readers or keys, or remove existing ones. Unique/shared device keys are not less secure but are problematic in terms of *(key) management*. This also means that P-Sec does not facilitate swift diagnosis and treatment in a secure manner during *emergencies*. This is because the paramedic R and I are likely unknown to each other and therefore do not share a K_{RI}. Since the type of messages between R and I are not symmetric (similar) in both directions, the protocol is not vulnerable to reflection attacks [93] **(SG-2.1.2.2)**. The use of the same key, or in other words, the lack of *key freshness* throughout all the communication sessions between R and I rules out key-confirmation attacks **(SG-2.1.2.3)** but creates replay-attack opportunities for A **(SG-2.1.2.4)**. In order to protect against these attacks, P-Sec employs nonces. This, however, depends on an error-free implementation of nonce genera-

tion. As a best practice, it is recommended to use different and unique keys for each session.

Moreover, when it comes to obtaining K_{RI}, it would be infeasible for A to find the 128-bit MISTY1 key using brute force **(SG-2.2.1)**. However, if A does manage to acquire the key, e.g., through social engineering **(SG-2.2.2)**, the protocol does not facilitate its replacement, and thus the only option would be to replace I through surgery.

SG-5: Denial of Service. Looking at SG-5, A can try DoS attacks by requesting a new session by sending a valid ID_R **(SG-5.1.2.1, SG-5.2.1.1)** (see Figure 2.5). After receiving N_I from I, A can send any message, which will force I to perform MAC calculations, resulting in battery drain. A can also disrupt the protocol and hence manage to cause function DoS by sending a bogus packet to I when it is waiting for a response from R, which will result in failed authentication and subsequently protocol reset **(SG-5.1.3)**. This will prevent the device from responding to legitimate requests.

2.4.2 P-KeyEx analysis

All the sub-goals of G-1 are relevant to the evaluation of P-KeyEx.

SG-1: Implementation Attack. In SG-1, we see that the protocol is independent of the need to install keys at manufacturing time and, hence, is free from resulting issues **(SG-1.1)**, as was the case with P-Sec. When it comes to flaws in protocol implementation **(SG-1.3.1)** we recognize that one possible candidate could be the improper design of the RNG that generates K_{RI}, however, it is not clear from [138] what algorithm is used to generate this random key. P-KeyEx does not allow distinguishing individual users (or user groups) for access control since it does not employ any pre-shared secret between R and I. Hence it is vulnerable to exploits targeting improper access control **(SG-1.4.2.1)**.

SG-2: Break Security Protocol. When evaluating against SG-2, we first analyze the symmetric cipher chosen for P-KeyEx, PRESENT-80 **(SG-2.1.1.1)**. The protocol uses a block size of 64 bits, however, birthday attacks are not a concern since the encryption process, i.e., $\mathsf{E}_{K_{RI}}(ID_I)$, uses a randomly generated key for every session. If we assume a strong internal RNG implementation, P-KeyEx protects against replay attacks **(SG-2.1.1.2, SG-2.1.2.4)**. In terms of key re-use **(SG-2.1.1.3)**, the protocol provides fresh keys for every session. The authors have done a comprehensive analysis of the randomness of biometrics (DCBs) employed in the protocol **(SG-2.1.1.4.1)**, hence there are no exploitable opportunities for the attacker in this regard. Usually when using ECCs **(SG-2.1.1.4.2)**, the effective key length is reduced since a portion of the key is used to provide redundancy for error correction, which sacrifices entropy [93]. However, in P-KeyEx the 80-bit K_{RI} has the same effective key length since it is encoded with 204 bits using BCH codes, which creates a Hamming distance of 37 bits between code words [138]. Similar to the reasoning

for P-Sec, P-KeyEx is not vulnerable to MITM and reflection attacks **(SG-2.1.2.1-2)**. Due to the key-confirmation steps at the end of the protocol, P-KeyEx is robust against key-confirmation attacks **(SG-2.1.2.3)**. The generation of fresh K_{RI} for every session protects P-KeyEx against brute-force attacks and makes side-channel attacks and social engineering inapplicable **(SG-2.2.1-3)**.

SG-3: Reader Proxy Attack & SG-4: Abusing Emergency Access. The only reader proxy attack applicable to P-KeyEx is the use of rPPG for heartbeat measurement **(SG-3.1.1)**. However, the high frame-rate requirement for the cameras, the need for the subject to be stable etc., make these attacks highly unlikely in practice. This *remote attack* is also the only relevant method for P-KeyEx in SG-4 **(SG-4.3.4.1.1)**.

SG-5: Denial of Service. Looking at SG-5, we see that P-KeyEx is susceptible to DoS attacks[7]. A can send ID_R to I to initiate a session **(SG-5.1.2.1, SG-5.2.1.1)**, and can also exchange valid misdetection flags to keep the session alive, even though he/she is not performing the witness generation his/herself (see Figure 2.6). This will force I to perform fuzzy commitment and hash calculation. A subsequent *msg* packet from A will result in an unnecessary decryption operation from I. Thus, A can cause serious battery drain using this method. A can also continuously modify or drop the misdetection flags during witness generation, which would result in R and I not being able to agree on IPIs needed for generating witnesses **(SG-5.1.3)**. Thus, A would be able to block any legitimate access (jam DoS).

2.4.3 P-Auth analysis

The evaluation of P-Auth has an attack-tree traversal similar to P-Sec and P-KeyEx. P-Auth shares issues pertaining to availability **(SG-5.1.2.1, SG-5.1.3, SG-5.2.1.1)** and access control **(SG-1.4.2.1)**. However, the major difference is the vulnerability to reflection attacks **(SG-2.1.2.2)** since the handshake is quite notably symmetric. A can exploit this by initiating connection with either R or I and then replaying the same messages that are received from either of these nodes after the session-key establishment. For instance, if A is trying to communicate with I, it can send its nonce, hash and IPI value equal to N_I, H_I and IPI_I, respectively. This would satisfy the checks for hash equality and IPI similarity at the final stage, resulting in incorrect authentication of A.

2.5 Recommendations

The results of the protocol evaluations in Section 2.4 are summarized in Table 2.2. Recall that here we have only shown the portion of the attack trees relevant to the evaluation. The impact of identified threats on the CIANA services is shown in Table 2.3, which also lists the coverage of the desired features specific to IMDs. Note

[7] It was not the objective of the authors of [138] to address availability.

Table 2.2: *Identified threats per protocol*

Sub-goal	P-Sec	P-KeyEx	P-Auth
1	SG-1.1, SG-1.3.1	SG-1.4.2.1	SG-1.4.2.1
2	SG-2.1.1.1, SG-2.1.1.2, SG-2.1.1.3		SG-2.1.2.2
3		SG-3.1.1	
4		SG-4.3.4.1.1	
5	SG-5.1.2.1, SG-5.1.3, SG-5.2.1.1	Same as P-Sec	Same as P-Sec

Table 2.3: *Evaluated-protocol services and additional features (N/E: non-eligible)*

Type of service/feature	P-Sec	P-KeyEx	P-Auth
Confidentiality	✓	N/E	N/E
Integrity	✓	N/E	N/E
Availability			
Non-repudiation			
Authentication	✓	✓	
Access control	✓		
Emergency access		✓	✓
Key management		✓	N/E
Key freshness		✓	N/E
Perfect forward secrecy		✓	N/E

that these protocols cannot be compared directly because of their different security aims. N/E stands for *non-eligible*, which denotes a service that was not intended to be supported by the protocol in question. As evident from the likelihood of attacks (see Figure 2.3), the protocols are most susceptible to DoS attacks, which hurt *availability*. We recognize that any similar protocol in isolation is susceptible to some form of DoS attack, however, in the case of IMDs, it is highly recommended to at least incorporate protection within the protocols against battery and function DoS, as evident from Section 2.2.4.1. One practical way to solve this issue is to use RF-energy-harvesting-based authentication [62, 164], which will be discussed in detail in Chapter 3. The protocols do not guarantee *non-repudiation* because they do not employ digital signature and public key infrastructure[8]. This is a valid concern since we assume the possibility of attacks from trusted entities (see SG-1.4 and G-2.3). Furthermore, P-Auth, in particular, fails to address *authentication* because of its vulnerability to reflection attacks. As an example, this can be resolved if both the nodes verify that the two nonces or IPI values are not exactly the same. Also, although P-KeyEx and P-Auth try to provide authentication in terms of establishing trust, they lack role identification/distinction, hence failing to address *access con-*

[8]IMDs normally do not have energy and computing resources to support asymmetric cryptography. Moreover, supporting non-repudiation also requires a robust logging infrastructure, which cannot be supported by a limited on-device memory.

trol. P-Sec can, however, support this feature since it facilitates multiple pre-shared passwords, allowing for multiple users (or user groups) to be distinguished. It can for instance appropriately utilize certain bits of ID_R for privilege information. It is not possible for A to modify these bits since ID_R is already pre-installed in I.

It can also be observed from Table 2.3 that the protocols have a largely non-overlapping coverage of the targeted features. If P-KeyEx is combined carefully with P-Sec to provide authenticated key exchange, the resulting scheme resolves P-Sec's issues related to symmetric-key usage and supplements it with emergency access. Moreover, in order to provide long-term *security*, some simple modifications are recommended: The block size employed for the PRESENT-80 cipher can be changed from 64 to 128 bits (assuming MISTY1 is not used because of the reasons highlighted in Section 2.4) to protect against birthday attacks. If this is not possible due to energy constraints, at least the session key should be changed frequently. In Chapter 4, we will discuss IMDfence, a security protocol for IMD ecosystems that provides a comprehensive yet practical coverage of the above security requirements.

It can be seen from the above discussion that attack trees can provide a very handy tool for quantifying weaknesses of the IMD systems. What is more, with an increasing volume of literature proposing security protocols for IMDs and related fields, employing attack trees can help to compile such contributions into new powerful protocols with a larger coverage of attacks, as demonstrated in the above evaluation of example protocols. The attack trees can be made more focused if the information about the commercial Reader/IMD-system implementations is made available. In essence, this work highlights the top-down approach instead of evaluating actual implementations.

2.6 Related work

Attack trees have been used as a tool to illustrate the attack scenarios within various domains and systems but their usage in the medical domain has been very limited. To the best of our knowledge there is no such work that specifically targets IMDs. Taylor et al. [167] have formulated an attack tree specifically applicable to the patient-controlled analgesia application, and subsequently suggested mitigating solutions. This work was extended by Xu et al. [184] who discuss a methodology to generate these attack trees. Luckett et al. [88] discuss the use of attack graphs for vulnerability identification, risk assessment and subsequent derivation of mitigation strategies to protect ambulatory medical devices (AMBs).

Populating the attack trees in this chapter has been based on in-house endeavor. Additionally, these were carefully expanded by consulting the following recent work in literature. Humayed et al. [68] discuss threats, vulnerabilities, attacks and security challenges of cyber-physical systems including medical devices. ALTawy et al. [4] study the trade-offs between security, safety and availability in cyber physical systems using IMDs as a case study. Camara et al. [26] survey the security goals for

future IMDs and analyze the protection mechanisms discussed so far in literature. Rathore et al. [125] provide an overview of the attacks pertaining to IMDs. Rushanan et al. [132] have done a rigorous survey of security schemes and attacks pertaining to IMDs and health-related BANs, and highlight emerging threats, but there is a need to perform a similar type of analysis to cover new attacks (e.g., [31, 94]) and protocols (e.g., [118, 138]), since the work was done more than seven years ago. Marin et al. [93] evaluate security of two physiological signal based security protocols for IMDs and have proposed solutions to improve security of such systems. Based on our work, we envision an open-access attack-tree resource where current research efforts can reflect upon and also contribute to.

2.7 Summary

In this chapter, we have proposed a systematic threat-modeling approach to analyze IMD security. This attack-tree-based approach offers a comprehensive and highly structured picture of the strengths and weaknesses of the IMD systems. As a case study, we applied our threat analysis on three IMD secure-communication protocols found in literature. We have showed that this evaluation makes the task of coming up with security improvements significantly easier. By using our approach, we have not only confirmed the capabilities/limitations of the protocols (as identified by their authors) but also discovered certain limitations (e.g., susceptibility to DoS and reflection attacks etc.). Moreover, it has enabled us to easily visualize and propose a combined use of these protocols, for better coverage of the identified security services and features. This work, thus, paves the way for building more robust and secure protocols for IMDs and mobile-health systems. What is more, it provides a structured approach towards performing system-level security evaluation to include many possible attack surfaces. We hope that this effort is a step in the right direction, towards the much needed standardization of IMD security.

CHAPTER 3

"A denial of service attack is as predictable for a site like this as the rain will fall one day or the sun will come up in the morning."

Australian Prime Minister Malcolm Turnbull
(after a denial-of-service attack on the census website), 11 August 2016

Towards realistic protection against battery-depletion attacks

M. A. Siddiqi, W. A. Serdijn, and C. Strydis, "Zero-Power Defense Done Right: Shielding IMDs from Battery-Depletion Attacks," *Journal of Signal Processing Systems,* pp. 1–17, 2020.

M. A. Siddiqi and C. Strydis, "Towards Realistic Battery-DoS Protection of Implantable Medical Devices," in *Proceedings of the 16th ACM International Conference on Computing Frontiers.* ACM, 2019, pp. 42–49.

In the previous chapter, it was shown that the *battery-depletion* (or *battery-DoS*) attack is one of the easiest to mount and highly effective attacks. This is also backed by the majority of the ethical-hacking efforts in which the batteries of commercial IMDs were depleted using black-box approaches [62,94]. In battery DoS, an attacker can force the IMD to continuously run an energy-consuming operation (such as authentication), which ultimately results in power loss and IMD shutdown.

It is considered that the only robust way of protecting an IMD against a battery DoS is by running the authentication operation using only *free* harvested energy. It can be argued that there is no necessity for this zero-power defense (ZPD) mechanism since technology exists to wirelessly charge IMD batteries when they are running low (as discussed in Section 3.1, next). However, this recharging feature is only available in less critical IMDs, such as spinal-cord stimulators [1, 104]. For critical devices such as pacemakers, there is a reluctance among the medical community to give recharging responsibility to the patients, in order to avoid patient errors. Moreover, the physicians prefer to replace the whole IMD after a certain period to get the latest technology [81]. Besides, even by assuming that all IMDs have this capability, the attacker can still drain the battery before the patient or doctor has a chance to recharge it.

Energy harvesting (EH) is a widely used concept employed in a variety of devices including RFIDs. However, ZPD for IMDs introduces new challenges that do not apply in other domains. Even though there are quite a few ZPD implementations proposed in literature, to the best of our knowledge the work described in this chapter is the first to facilitate the transition from *concept* to *industry-compliant* ZPD designs for IMDs. Based on a clear-cut set of design considerations, we survey and evaluate the current state of the art and proceed to propose specific recommendations for enhancing existing IMDs. Essentially, this chapter makes the following novel contributions:

- We consolidate ZPD design considerations for the specific domain of IMDs.
- We perform a survey of existing systems and highlight their limitations based on the above considerations.
- We provide recommendations in order to develop comprehensive protection of IMDs against battery-DoS attacks.

The rest of the chapter is organized as follows. We provide a brief background on the use of energy harvesting in IMDs in Section 3.1, and then provide motivation for using it to enhance IMD security in Section 3.2. In Section 3.3, we provide detailed ZPD design considerations. Based on these considerations, we review and evaluate state-of-the-art ZPD solutions in Section 3.4. In Section 3.5, we provide recommendations for improving ZPD designs. We conclude the discussion in Section 3.6.

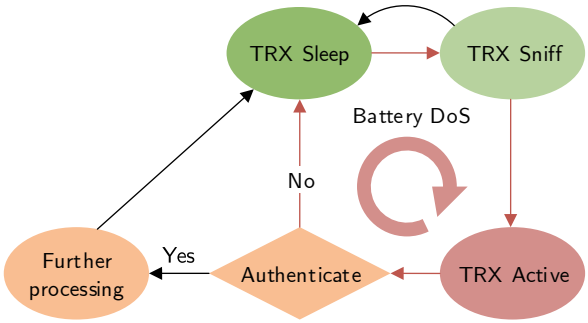


Figure 3.1: *Battery-DoS attack: continuous traversal of the different transceiver modes and the authentication protocol*

3.1 Energy harvesting in IMDs

The use of energy harvesting in IMDs is not new. The application of this concept, however, has been very narrow in this domain, i.e., in wireless power transfer (WPT)[1] to recharge IMD batteries. For instance, there are several rechargeable neurostimulators that are commercially available [1, 104]. In this specific category of implants there is a rising trend towards increased IMD-power requirements due to recent advances in neuromodulation-related pain relief. For such power-hungry devices, a non-rechargeable battery would result in a very short IMD lifespan and subsequently require expensive surgeries in order to replace the battery-depleted implants. One way of avoiding this is to use larger battery sizes, which can quickly become impractical to implant. Hence, the natural solution is to use rechargeable systems, which can prevent the need for frequent surgeries and would result in smaller battery sizes and implants as a whole [106].

3.2 Energy harvesting for battery-DoS protection

During normal IMD operation, the RF transceiver usually polls for an external entity by cycling through *sleep* and *sniff* modes [109]. The (short-duration) sniff mode consumes relatively little power compared to active transceiver operation. If the transceiver detects RF energy, it switches to its active mode in order to receive data. Battery-DoS attacks basically change the sleep-awake periods of both the IMD transceiver and the internal processor, as shown in Figure 3.1.

Battery-DoS attacks can generally happen in two ways [55]: (1) They can increase the IMD activity by sending bogus communication packets. As an example,

[1]The term *energy harvesting* generally refers to harvesting energy from ambient sources, whereas *WPT* refers to the intentional transfer of energy from a dedicated charging device [34]. In this chapter, we use the terms interchangeably.

the attacker can repeatedly request the IMD to establish a secure channel using incorrect credentials. Consequently, the IMD will run part of an energy-consuming authentication protocol for analyzing every request, which will drain the battery. (2) The attacker can also generate electromagnetic (EM) noise in order to cause high error rates at the IMD transceiver, which in turn increases its energy consumption due to increased number of retransmissions. This increased noise may also force the IMD to increase the transmission power, which also reduces battery life.

In light of the fact that energy harvesting has already been employed by some classes of IMDs, the use of this concept, in the form of ZPD, has now become quintessential to protecting all IMDs against battery DoS. In this scheme, the IMD, while authenticating the external entity that is trying to communicate, can run the energy-consuming security primitives using the RF energy harvested from the incoming communication messages. The IMD is allowed to use the battery for subsequent operations *only* after the entity is authenticated. This prevents the IMD from depleting its battery to entertain continuous bogus messages from a malicious entity.

3.3 Design considerations

In this section, we enumerate and discuss various considerations that should be taken into account when approaching the design of an IMD-specific ZPD system.

3.3.1 Choice of WPT technique

Since ZPD is based on the concept of wireless energy harvesting, it is important to briefly discuss the WPT techniques that enable such strategies. A typical WPT setup is shown in Figure 3.2 [75, 89]. State-of-the-art IMD-specific WPT techniques can be broadly categorized into three types[2] [14]:

3.3.1.1 Inductive Coupling (IC)

Near-field or magnetostatic WPT is usually categorized as inductive coupling or inductive power transfer (IPT). IPT usually involves the use of two coupled coils that have the same inductance. The transmitter coil is placed outside the body. When an AC current passes through it, voltage is induced due to electromagnetic induction in the receiver coil, which is located inside the body. IPT is the dominant method that is used to wirelessly recharge commercial IMDs, specifically neurostimulators [1, 104].

3.3.1.2 Radio Frequency (RF)

If the transfer is in the transition region (mid field) [66] or far field, then the WPT system is usually categorized as RF or electromagnetic power transfer (RFPT). Here,

[2]Note that this classification is not universal.

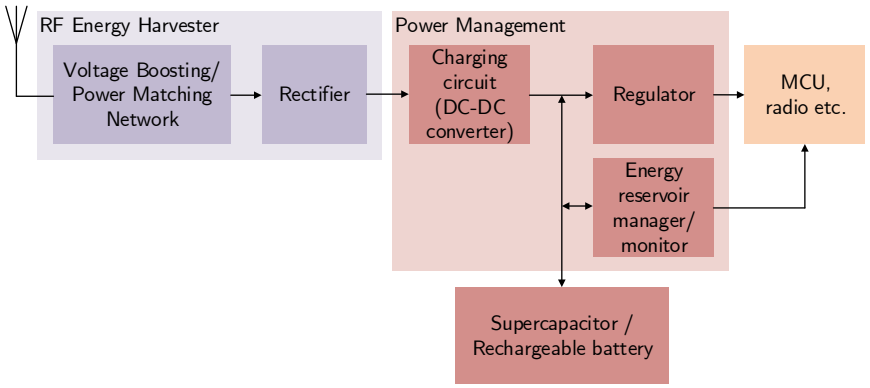


Figure 3.2: *A typical WPT System (RF Energy Harvesting)*

Table 3.1: *Comparison of WPT techniques*

Technique	Range	Biological effects	Transferred power	Receiver size
IPT	−	−	+	−
RFPT	+	−	−	+
APT	−*	+	+	+

+/−: relatively good/poor performance, *: requires (non-air) medium

antennas are not just limited to coils for the transmission of power. A typical RFPT system is shown in Figure 3.2.

3.3.1.3 Acoustic/Ultrasound

This WPT category harvests acoustic waves, which are usually at ultrasound frequencies. In acoustic power transfer (APT), the transmitter node, while in contact with the skin, generates these waves using a piezoelectric transducer. These waves induce charge differences on a piezoelectric device in the receiver node, which is located inside the body along with the IMD.

The advantages and drawbacks of the three WPT techniques are summarized in Table 3.1 in terms of operating range, potential biological effects, amount of transferred power and receiver area. The choice of WPT scheme and associated transferred-power amount has an impact on the real-time IMD *performance*, and also on the *size* of the energy reservoir and, subsequently, the IMD as a whole. This is further discussed in the subsequent sections.

3.3.2 Medical-safety constraints

The ZPD technique should satisfy the various requirements by the FDA, FCC, IEEE, etc., in order to prevent any adverse biological effects on human tissue due to excess electromagnetic-energy exposure. IEEE puts constraints on the intensity of RF signals and defines maximum-permissible-exposure (MPE) limits for magnetic and electric fields [70]. In addition to RF-signal intensity, the signal frequency has a significant impact on the amount of energy absorbed in the human tissue and the resulting potential to cause harm. This absorption is characterized by the *specific absorption rate* (SAR), which is expressed in $\frac{\text{W}}{\text{kg}}$ or $\frac{\text{mW}}{\text{kg}}$. The peak-spatial-average SAR values for exposure of the public and controlled environments are 2 $\frac{\text{W}}{\text{kg}}$ and 10 $\frac{\text{W}}{\text{kg}}$, respectively (over 10 g of tissue) [70]. The FDA also has guidelines regarding intensity of acoustic signals in $\frac{\text{W}}{\text{cm}^2}$, namely *spatial peak temporal average intensity* (I_{SPTA}) and *spatial peak pulse average intensity* (I_{SPPA}) [48]. Satisfying these constraints impacts the choice of WPT scheme (as discussed in Section 3.3.1).

3.3.3 Frequency-band constraints

Certain FCC constraints also need to be met in order to avoid IMD-radio interference with other devices operating in the same frequency band. For example, the MedRadio band, which is reserved for IMD communication, does not allow an equivalent isotropically radiated power (EIRP) of more than 25 μW [46]. Since this amount of power is too small for WPT (as will be discussed in Section 3.5.3), a separate band should be used for power transfer, whereas the MedRadio band can be used for data communication. This implies increased cost and size due to the use of two antennas. One solution could be to use a single ISM-band (13.56 MHz) antenna for both WPT and data communication, however this would result in lower data rates due to smaller allowed bandwidth than that of MedRadio [98].

3.3.4 Real-time behavior

Harvested power needs to stay above the consumed power in order for the energy consumers to work seamlessly. Otherwise, an energy reservoir must be employed so as to collect sufficient energy before the IMD can use it. Technically, due to this reservoir, the ZPD scheme should always work, but the charging delay limits usability and real-time behavior, which can be critical in the case of emergencies. The ZPD scheme should never slow down a paramedic access and jeopardize patient safety as a result.

3.3.5 Choice of energy reservoir

Either a supercapacitor (supercap) or a rechargeable battery can be employed as the energy reservoir. Supercaps in general have a longer lifespan and support more recharge cycles than batteries [91], and thus are more suitable for IMDs. Employing

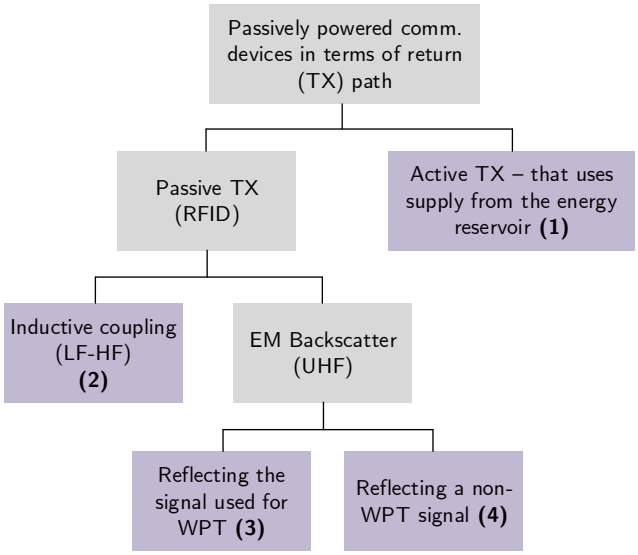
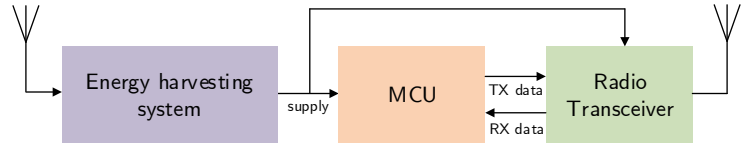


Figure 3.3: *Classification of passive communication devices in terms of transmitter implementation*

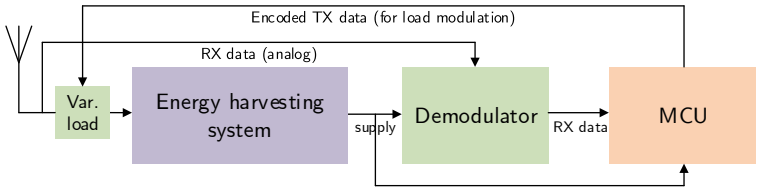
a supercap, can limit the range of applied charging voltage, since these components have low operating-voltage limits. Also, as indicated in [152], the capacitor size has to incorporate the losses due to the decoupling capacitors connected to the energy consumers.

3.3.6 Passive wireless communication

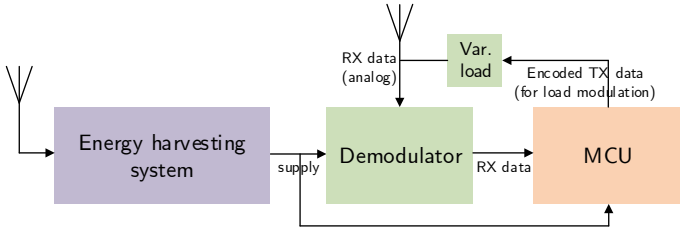
Passive communication relies on WPT schemes in order to function without the need of an on-board power supply. This concept forms the basis of ZPD strategies, which will be discussed in Section 3.4. The most critical component of these passive devices is the wireless transceiver that can lead to significant peak power consumption based on the design choice. Based on the choice of the transmitter, which subsequently impacts the receiver implementation, we categorize these devices into four schemes, as depicted in Figure 3.3. The different schemes at the leaf nodes of the tree are numbered accordingly and are subsequently explained. The first part of the scheme name indicates the type of wireless communication whereas the suffix indicates whether the communication shares the power-transfer-signal frequency band (PB) or uses an independent band (IB).



(a) ActiveTX-IB



(b) IC-PB and EMB-PB



(c) EMB-IB

Figure 3.4: *Schematics of different passive communication schemes for ZPD*

3.3.6.1 ActiveTX-IB

The passive device has an *active* transceiver, i.e., it actively transmits (using supply from the energy reservoir) instead of reflecting the incident RF signal, as shown in Figure 3.4a. This scheme is employed by the design in [136].

3.3.6.2 IC-PB

The downlink (reader to passive device) communication uses the same signal that is used for inductive power transfer, which lies in the low- or high-frequency band (LF-HF). For the uplink, the electrical properties of the inductive coil are changed (by load modulation; in this case, *Load Shift Keying*), which affects the same inductive-coupling field, and is thus detected by the reader (see Figure 3.4b). The design in [27] employs this scheme.

3.3.6.3 EMB-PB

Compared to the previous scheme, RF/Electromagnetic backscattering (EMB), which reflects the incident RF, is used for data transmission instead of inductive coupling.

Here, the incident RF is used for both energy harvesting and data communication (see Figure 3.4b). The RF is reflected if the load across the antenna feed-point is minimum, and vice versa. One of the works that employ this scheme is [133]. The use of EMB helps eliminate the high peak power consumption of a conventional RF transmitter. This is important for passive devices because, even to transmit just a few bits of data, the peak power may exceed the incoming power, which will result in device malfunction in the absence of a reservoir. Note that the use of EMB for transmission is fully beneficial only if a simple and low-power circuit is used for the receive path, such as an Amplitude-Shift-Keying (ASK) envelope detector.

3.3.6.4 EMB-IB

Compared to EMB-PB, here the difference is that the WPT signal is different from the one used for EMB (as shown in Figure 3.4c). The design in [89] uses this scheme.

ActiveTX-IB and EMB-IB offer the most flexibility since they use separate antennas for WPT and data communication. As discussed in Section 3.3.3, these configurations are helpful in meeting the FCC constraints while maintaining both the sufficient power transfer and data rates. On the other hand, IC-PB and EMB-PB are more economical in terms of resources since they only employ one antenna [98]. This comes, however, at the cost of reduced flexibility in terms of data rate.

3.3.7 Fundamental security services

ZPD schemes primarily address *Availability* from the CIANA security services (see Section 2.2.3). Ensuring the rest of the services can have an indirect impact on Availability. As an example, if the IMD has a dedicated processor that is responsible for *authenticating* an external entity, the peak-power consumption of the implant will increase when this peripheral is active. As a result, the bogus messages sent by an attacker will draw more energy from the battery than in the case of a less-secure IMD. Hence, ensuring one service should not be at the expense of the other.

The choice of cryptographic primitives, which are needed to provide these services, plays a critical role in the design of the energy-harvesting circuit. For example, lightweight block ciphers are preferred candidates for achieving data confidentiality because of their low energy profile. Moreover, in order to achieve integrity and authentication, a cipher-based Message Authentication Code (MAC) should be used instead of a hash-based MAC (HMAC) because of lower energy consumption in software implementations. For dedicated hardware implementations, however, this is not always the case [117]. Furthermore, for these systems, *mutual* authentication should be employed instead of just authenticating the reader unilaterally. This is required to prevent spoofing attacks on the reader [164]. This implies that the harvested energy should be able to support *both* transmission and reception of data.

In addition to battery DoS, satisfying availability also implies providing protection against a second class of DoS attack on IMDs, which we call function DoS [164].

This type of DoS floods the IMD with communication requests in order to prevent it in performing its main medical functionality. It follows that a security architecture that ensures implant availability should protect against both battery and function DoS.

3.3.8 Device usability

The ZPD design should not result in an awkward usage or programming of the IMD. For instance, a (short range) IPT-based access does not result in patient inconvenience in case of a pacemaker. However, such a close-range access may not be acceptable to the patient in case of a neurostimulator implanted within the skull. For instance, a patient may avoid frequent IMD access in public if it requires placing the reader interface on their head. This can significantly impact the patient's social life and can lead to social segregation.

3.3.9 Maintainability

The ZPD design involves executing cryptographic mechanisms in order to authenticate the external entities. In the event of discovering a security loophole or an implementation bug, the cryptographic primitives and/or the security protocol require replacement or an update. In the case of IMDs, the first requirement is that such updates should be possible at the firmware level using the wireless interface, otherwise hardware updates or modifications imply device explantation via surgery. Another requirement is that such firmwares should be decoupled from the main IMD functionality. This expedites the firmware-certification cycle. Updating a monolithic IMD firmware, which includes both the medical application and the security functionality, is highly likely to result in a longer certification cycle compared to a decoupled firmware.

3.3.10 Reliability

IMDs are safety-critical systems, which have extreme safety and reliability constraints. A ZPD design introduces additional electronic components to the IMD system, and each component (e.g., a transistor) has an associated failure rate. Hence, ZPD protection should not significantly impact the overall implant reliability. Analysis of IMD reliability is further discussed in Section 3.5.6.

3.3.11 Emergency access

In the case of emergencies, the paramedics or first responders should have unhindered and fast access to the IMD, without compromising patient safety and security. Hence, an appropriate balance should be attained between usability, safety and security. For instance, the authentication protocol running on harvested energy should not require a *pre-shared* secret (or key) between the reader and the IMD. Other-

wise, it will not work in the case of the paramedic reader, which will most likely not have the same key as the implant. Moreover, as discussed in Section 3.3.4, it is of paramount importance that the choice of WPT and the associated energy reservoir results in acceptable charging delay in order to ensure real-time performance. Otherwise, it will block legitimate access to the IMD in emergency scenarios.

3.3.12 Design suitability

Existing IMD designs take a long time from concept to market due to rigorous certification cycles. Therefore, any new ZPD solution should fit in seamlessly in the existing designs resulting in minimal changes and short review cycles. For example, technically speaking, a large energy reservoir enables ZPD but this increases the size of the design and introduces unnecessary delay, which impacts suitability.

3.3.13 Conformity to touch-to-access principle

Any ZPD scheme shall ensure that only the entity in close proximity to the patient for a prolonged period of time is allowed to access the IMD (see Section 2.2.5.1).

3.3.14 Range of operation

The ZPD solution shall be able to work correctly independently of the implantation depth. Appropriate balance should be attained between the WPT and the associated thermal effects and energy absorption in the human tissue. Also, the ZPD solution shall allow the provision of bedside-base-station operation for the convenience of the patient (see Figure 1.2). This device by definition can be less than 10 feet away from the patient [159]. However, in order to conform to the touch-to-access principle, this communication should be strictly limited to the bedside range (less than 5 feet away).

3.4 A survey of existing ZPD techniques

In light of the design considerations discussed in Section 3.3, we now survey works from literature and discuss their limitations. We hope that this survey will help us reflect on and validate the design considerations. These works are presented in chronological order, and to the best of our knowledge, are the only works pertaining to ZPD for IMDs.

3.4.1 Harvesting-based techniques

Halperin et al. [62] presented the pioneering work of RFID-style energy harvesting for zero-power defense of IMDs. They use an RFID module called WISP [133], which employs EMB for the data transmission from the implant to the reader, and simple ASK-envelop detection in the reverse direction, while using RFPT for wireless power transfer. Their scheme, however, does not perform mutual authentica-

tion and its acoustic-communication-based key transport is susceptible to attacks, as shown in [61].

The scheme from Liu et al. [87] is the only ZPD work that takes FCC regulations into consideration. They employ the ISM band for RFPT and the MedRadio band for data communication. It employs a dedicated passive RFID wake-up module, which performs RF-energy harvesting from the incoming signal in order to authenticate the other entity. Upon successful authentication, the main module is woken up. This scheme uses pre-shared keys between the reader and the IMD, which makes emergency access impossible. This is because in emergencies, the IMD and the paramedic reader are likely unknown to each other and therefore do not share a key.

Strydis et al. [164] propose an IMD architecture that isolates the implant functionality from the security tasks by using dedicated processing cores for the respective applications. They designed the security co-processor from scratch, which was optimized for executing the MISTY1 cipher in terms of energy and performance. The choice of this dual-core architecture helps in dealing with repeated communication requests that may prevent the implant from performing its primary task. Thus it effectively protects against function DoS. Battery DoS is tackled by ensuring that the security core and the transceiver run on harvested RF energy before mutual authentication of reader/IMD. After successful authentication, these modules are allowed to use battery power for subsequent communication. However, they did not present a full system implementation.

Ellouze et al. [42, 43] propose an RFID-based, energy-harvesting solution, that uses the same WISP module as employed by [62]. In contrast to [62], their solution additionally provides mutual authentication. They use cardiac-signal-based biometrics for authentication and the generation of session keys. However, the fuzzy-vault-inspired protocol (OPFKA) [67] employed in their scheme is vulnerable to attacks as demonstrated in [130].

Yang et al. [186] use IPT, and employ the same coil for power transfer and data communication. Their scheme provides mutual authentication. However, it employs pre-shared keys, and is thus unable to support emergency access. Moreover, they did not implement a unified ZPD system since the hash-based authentication was verified separately on an FPGA.

Chang et al. [27] propose a generic ZPD solution that is not specific to IMDs per se, however, it covers a spectrum of devices that have more or less the same profile. They propose IPT for the power transfer from the reader. This signal is also used for bi-directional communication. However, they do not give any description of the employed security protocol.

3.4.2 Non-harvesting-based techniques

Denning et al. [38] propose a class of defensive mechanisms, which uses an external device, called the communication *cloaker*. This device shares a secret key

with the IMD, which allows secure communication between the pair. The defensive mechanisms vary in terms of whether the IMD checks the presence of a cloaker *periodically* or if it contacts the cloaker only when an external entity tries to access the implant. In case the cloaker is absent, the IMD allows fail-open access to any reader. Otherwise, the cloaker performs the authentication of the external entity, and allows it to communicate with the IMD in case it is authentic. Although the proposed class provides emergency access, the authors acknowledge that it is susceptible to jamming attacks, in which the attacker selectively jams packets between the cloaker-IMD pair in order to convince the IMD of the cloaker's absence. Additional mitigation schemes against these attacks are briefly discussed. Another drawback of this scheme is that it introduces an additional single point of failure. This is because the IMD becomes unsecured in case the patient forgets to wear the cloaker, or loses it.

Hei et al. [64] utilize the concept of *anomaly detection* [132] in which the system automatically detects abnormal events, such as malicious access. Their scheme is based on supervised learning in which the normal access patterns of IMDs are used as training data. The result is then used to classify abnormal IMD accesses in real time. Their scheme uses an additional device (a cellphone) that performs this real-time classification. Moreover, their scheme is designed to block anomalous access attempts *before* the expensive authentication-related computations are performed by the IMD. When the IMD is contacted by an external device, it asks the cellphone to classify this connection attempt. Based on the verdict from the cellphone, the IMD either proceeds with the authentication, or goes to sleep. One main drawback of their scheme is that they have neither provided a security protocol between the IMD and the cellphone, nor any security analysis. One highly probable attack against this scheme is for an attacker to spoof cellphone messages to the IMD. Moreover, this scheme is not designed to work in an emergency scenario.

Similarly to [38], Gollakota et al. [58] propose an external wearable device, called the *shield*, which listens and jams all IMD accesses. With this *friendly* jamming, the scheme tries to protect against both active and passive (eavesdropping) attacks. In case a legitimate reader access is required, the shield is simply removed from the patient's proximity. The main advantage of this solution is that it can be readily employed in existing IMD systems. However, similar to [38], this scheme introduces an additional single point of failure. Moreover, they assume that the distance between the IMD and the shield is less than the distance between the attacker and the IMD, and hence the attacker would be unable to perform eavesdrop attack. However, it is shown in [171] that MIMO-based eavesdropping attacks are possible if the attacker uses two antennas within 3 meters of the patient [132].

Table 3.2: *Summary of ZPD strategies*

Design considerations	Harvesting-based techniques						Non-Harvesting-based techniques		
	Halperin et al. [62]	Liu et al. [87]	Strydis et al. [164]	Ellouze et al. [42, 43]	Yang et al. [186]	Chang et al. [27]	Denning et al. [38]	Hei et al. [64]	Gollakota et al. [58]
Satisfy safety constraints	-	-	-	-	-	-	N/A	N/A	N/A
Satisfy freq. band req.	-	Yes	-	Yes	Yes	-	Yes	Yes	Yes
Real-time performance	Yes	-	-	-	-	Yes	-	-	Yes
Energy reservoir	Not used	Not used	-	Not used	Not used	-	N/A	N/A	N/A
Type of WPT technique	RFPT	RFPT (ISM)	-	RFPT	IPT	IPT	N/A	N/A	N/A
Passive wireless comm.:							N/A	N/A	N/A
Scheme	EMB-PB	-	-	EMB-PB	IC-PB	IC-PB			
ZPD Receive path	ASK	-	-	ASK	IC (ASK)	IC (FM)			
ZPD Transmit path	EMB	No	-	EMB	IC (ASK)	IC (ASK)			
Security services related:									
Employed primitives	DE	-	DE, CMAC	DE, HMAC	Hash func.	DE	-	-	None
Mutual authentication	No	No	Yes	Yes	Yes	-	No	No	No
Avoids pre-shared keys	Yes	No	No	Yes	No	-	Yes	-	Yes
Function-DoS protection	Yes	Yes	Yes	Yes	Yes	-	-	-	Yes
vulnerabilities	[61]	-	-	[130]	-	-	SJ	-	[171]
Device usability	-	-	-	-	-	-	-	-	Yes
Reliability	-	-	-	-	-	-	SPF	SPF	SPF
Maintainability	Yes	-	Yes	Yes	No	-	-	-	Yes
Emergency access	Yes	No	No	Yes	No	-	Yes	No	Yes
Touch-to-access	Yes	-	No	Yes	-	Yes	Yes	No	Yes
Operating distance:									
Max. implantation depth	1 cm (animal tissue)	-	-	-	-	2.5 cm (air)	N/A	N/A	N/A
Bedside-base-station	No	No	No	No	No	No	No	-	No
Design suitability (size, additional hardware etc.)	WISP and piezoelectric transd.	RFID module	Co-proc. for security	WISP	-	-	ED	ED	ED

'-': Lacking information, DE: Data Encryption, Not used: Advantageous avoidance of an additional component, Yes/No: Satisfies requirement (or not),
N/A: Not applicable, ED: The scheme employs an external mobile/wearable device, SJ: Selective-jamming attack, SPF: ZPD design adds an additional single point of failure

3.4.3 Summary

Table 3.2 compares the above ZPD techniques based on the various parameters and design considerations highlighted in Section 3.3. We can see that all harvesting-based works lack the evaluation of hazardous biological effects of the employed WPT schemes. Moreover, all the techniques do not consider the possibility of a bedside-base-station operation, which is a rising trend in the reader/IMD systems. They also offer insufficient security services and/or have security vulnerabilities in one form or another.

3.5 Discussion and recommendations

We, next, provide recommendations on how existing solutions can be improved in order to better meet the design constraints highlighted in Section 3.3.

3.5.1 Adaptive ZPD

In modern IMD setups, in addition to the doctor's programmer, we also have a bedside base-station, as shown in Figure 1.2. For the convenience of the patients, these wireless devices are required to communicate with the IMD from a few feet away [159]. With this constraint, IPT- and APT-based ZPD cannot be used for the base-station/IMD authentication. Hence, with this setup, it is advantageous to employ RFPT for energy harvesting, since it is more flexible compared to IPT and APT in terms of range. Though the amount of power transferred through RFPT is significantly smaller compared to IPT/APT, it is not an issue in this specific case since the base-station communication is only used for non-critical daily monitoring. As a result, this setup can afford long delays due to energy-reservoir charging. In light of the above, an *adaptive ZPD* approach should be considered, that e.g., uses IPT/APT for doctor-programmer/IMD communication, and switches to RFPT for base-station/IMD communication. In terms of implementation cost, it is more economical to use IPT for programmer/IMD communication instead of APT. This is because the same coils can potentially be employed for near-field (programmer communication) and far-field (base-station communication). On the other hand, the use of APT (for programmer communication) would require the use of piezoelectric transducers in addition to the RF antenna (needed for base-station communication).

3.5.2 Main-implant-battery size

We now discuss how realistic it is to achieve battery DoS when considering actual IMD battery sizes. The generic components of the total IMD energy consumption are summarized in (3.1) [39].

$$E_{total} = E_{comp} + E_{sense} + E_{stim} + E_{TRX} \tag{3.1}$$

Table 3.3: *Specifications of a typical pacemaker*

Parameter	Value
Supply voltage	3.3 V
MCU-processor clock frequency	19 MHz (default value) [153]
RF-transceiver effective data rate	265 kbps (maximum value) [109]
Active-data-comm. duration	3 minutes per day [159]
Pacemaker stimulation energy	20 μJ per heartbeat [39]
Medical-application duty cycle	5% [86]

E_{comp} is the computational energy which includes the energy spent by the IMD processor or MCU for medical-related processing, and the energy spent for handling the incoming or outgoing communication messages. E_{sense} is the energy consumed during the sensing of a physiological signal from the human body. E_{stim} is the energy spent for electrical stimulation via the electrodes applied by the IMD on the human tissue. Finally, E_{TRX} is the energy consumed by the RF transceiver.

For the calculations, it is assumed that the IMD has a state-of-the-art ultra-low-power ARM Cortex-M0+ based 32-bit MCU [153], running at 19 MHz, and an implantable-grade radio transceiver [109], with an effective data rate of 265 kbps. The supply voltage is set at 3.3 V. Moreover, the worst-case E_{stim} of a pacemaker is 20 μJ per heartbeat, based on reported figures of commercial devices [39]. The duty cycle of the transceiver is estimated at 0.21%, which corresponds to 3 minutes of active data communication per 24 hours with a bedside base-station [159]. The above parameters are summarized in Table 3.3. The IMD-battery-lifetime trends with respect to example processor duty cycles, which contribute to E_{comp} and E_{sense}, are shown in Figure 3.5. For instance, the pacemaker design in [86] has a processor duty cycle of 5%. The data points correspond to actual implantable-grade battery sizes [40].

The time required to completely deplete the IMD battery by continuously sending bogus communication packets is illustrated in Figure 3.6. On average, we assume half the charge available in the batteries due to normal use. As a worst-case scenario, we also assume that the authentication steps are executed *continuously* on *active* modes of the MCU and the transceiver with the current consumption of 0.78 mA and 4.9 mA, respectively. It can be deduced from these plots that, as a first level of defense, the battery sizes for critical applications, such as pacemakers, should be as large as possible.

We now analyze the effect of the EM-noise attack, in which the attacker's aim is to cause IMD retransmissions due to high error rates at the IMD transceiver. Based on the analysis from Gelenbe et al. [55] of battery-DoS attacks on sensor nodes, the IMD current consumption under an EM-noise attack can be represented by (3.2).

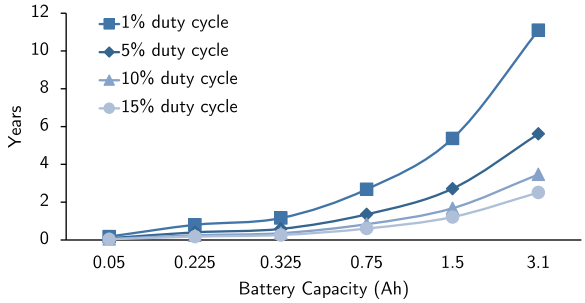


Figure 3.5: *IMD-battery lifetime with respect to example processor duty cycles while the transceiver is active for 3 minutes per 24 hours*

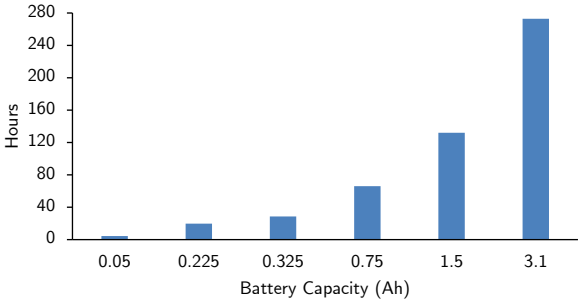


Figure 3.6: *Time required to completely deplete a half-full IMD battery through battery DoS*

$$I_a = I_n + rI_a = \frac{I_n}{1-r} \tag{3.2}$$

Here, I_n is the average current consumption in a normal scenario, I_a is the total average current consumption in an attack scenario, and r is the retransmission probability ($0 \leq r < 1$). Increase in the EM-noise level is reflected by an increase in r. Figure 3.7 shows the expected lifetime of an IMD operating under realistic processor and transceiver duty cycles of 5% and 0.21% respectively (as discussed above). From this, we can conclude that, although the EM-noise attack significantly affects the IMD lifetime, its impact is less critical compared to continuously making bogus authentication attempts. This is because the amount of RF traffic generated by the IMDs in realistic scenarios is very low, e.g., 3 minutes per day for the above-mentioned reader [159].

3.5.3 Reservoir size and charging delay

If the peak power of the load is always less than the harvested power, then we do not need a reservoir. Otherwise, the size of the reservoir is determined by looking

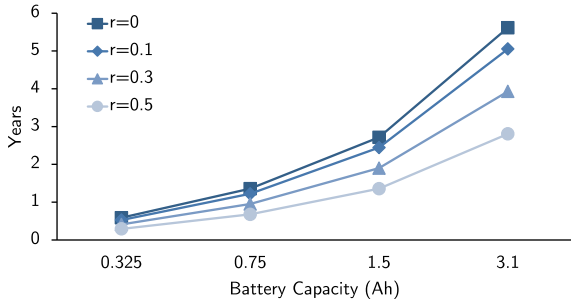


Figure 3.7: *IMD-battery lifetime in the presence of an EM-noise attack resulting in the retransmission probability r*

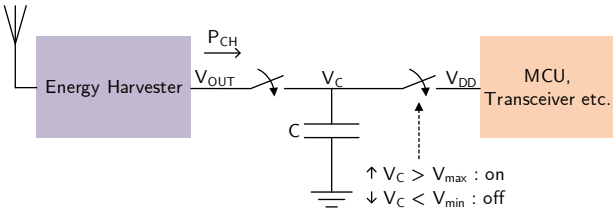


Figure 3.8: *Simple ZPD configuration*

at the required energy consumption of all the consumers during the authentication operation. Moreover, if a reservoir is required, then it may seem that any ZPD scheme might work. However, this is not true since it can become impractical for high-energy-consumption solutions due to the long delay, which is required to store sufficient energy.

For capacitor reservoirs, in order to determine the required capacitance, the energy in the capacitor *available* for use (ΔE), should be greater than the authentication energy (E_{auth}). The capacitance can be calculated using (3.3) [37], where V_{max} is the capacitor voltage when it is sufficiently charged and V_{min} is when it has been used by the application or authentication process (see Figure 3.8).

$$\Delta E = \frac{1}{2}\, C\, (V_{max}^2 - V_{min}^2) \;>\; E_{auth} \tag{3.3}$$

RF-energy harvesters in general output *constant power* instead of constant voltage [110]. In this type of capacitor charging, the supplied voltage increases (instead of staying fixed) and current decreases with increasing capacitor voltage. The capacitor charging time[3] (t_{ch}) for this type of charging is calculated using (3.4) [110]. Here, P_{ch} is the charging power supplied by the energy harvester to the capacitor

[3]The capacitor charging time for constant voltage charging is $5RC$.

(C), R is the capacitor's equivalent series resistance (ESR) and Q is the amount of coulombs stored during this time. Here $A = \sqrt{Q^2 + 4C^2RP_{ch}}$.

$$t_{ch} = \frac{Q^2 + QA + 4C^2RP_{ch}\ \ln(\frac{A+Q}{\sqrt{4C^2RP_{ch}}})}{4CP_{ch}} \tag{3.4}$$

If the authentication-energy consumption is reduced, then the required reservoir capacitance can be reduced as a result. If this value is between 0.1 μF and 470 μF, then ceramic capacitors can be employed, which are ideal for energy harvesting because of their low leakage current, small size and low cost [37]. These capacitors also have a very low ESR [45], which allows us to ignore the effect of the time constant (RC). Hence, (3.4) can be simplified to (3.5), which is also equivalent to (3.6). Here, E is the energy stored in the capacitor.

$$t_{ch} = \frac{Q^2}{2CP_{ch}} \tag{3.5}$$

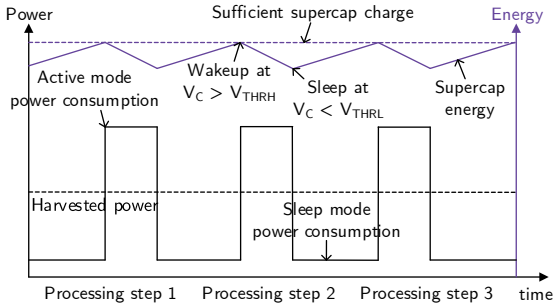
$$t_{ch} = \frac{E}{P_{ch}} \tag{3.6}$$

The time it takes to charge an empty capacitor ($t_{ch_{initial}}$), and in the case of subsequent charging operations ($t_{ch_{repeat}}$) when a capacitor has a residue voltage of V_{min} can be calculated by (3.7) [37]. Here, $E_{initial} = \frac{1}{2}CV_{max}^2$, which is the energy attained by an empty capacitor when charged from 0 V to V_{max}.

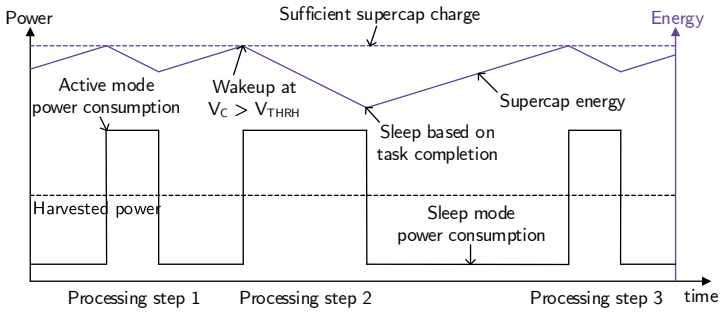
$$\begin{aligned} t_{ch_{initial}} &= \frac{E_{initial}}{P_{ch}} \\ t_{ch_{repeat}} &= \frac{\Delta E}{P_{ch}} \end{aligned} \tag{3.7}$$

As an example, we use the evaluation setup from Section 3.5.2 and take the ISO/IEC 9798-2-based mutual authentication protocol from the ZPD solution in [164]. We use AES-128 for data confidentiality and cipher-based MAC. For WPT, we look at the IPT scheme from [85], which is specifically designed for IMDs and delivers $P_{ch} = 6.15$ mW. Using $V_{max} = 3.3$ V and $V_{min} = 2.1$ V, which are within the operating supply voltage range of this setup (i.e., 2.05 V to 3.5 V), we see that C for the resulting scheme turns out to be 6.19 μF (since the measured $E_{auth} = 20.07$ μJ). Using a standard ceramic capacitor of size greater than this value e.g., 10 μF, $t_{ch_{initial}}$ and $t_{ch_{repeat}}$ turn out to be 8.85 ms and 5.27 ms respectively, which are very reasonable in terms of real-time behavior.

In general, the simplest solution is always to choose a reservoir capacitance that is much larger than the required value (as long as the charging delay is reasonable). This margin is important since the authentication protocol or the employed cryptographic primitives can change in the future, e.g., due to security updates. However,



***(a)** Entering sleep based on voltage-comparator interrupt*



***(b)** Entering sleep after protocol-step completion*

***Figure 3.9:** Supercapacitor characteristics in relation to application duty cycle (active mode vs. sleep mode)*

in case C turns out to be outside the ceramic-capacitor range due to large E_{auth}, we can employ the following schemes to reduce it, and thereby the charging delay.

3.5.3.1 Use of sleep modes

The capacitor-charging delay can be minimized by using sleep modes and interrupts, instead of sizing the capacitor for the whole authentication, resulting in reduced required capacitance. One way of achieving this could be to achieve a minimum required voltage (V_{THR_H}) using a voltage-controlled switch, before the capacitor energy is used by the rest of the IMD (Figure 3.9a). After some processing, the implant MCU can then enter sleep mode based on a voltage-comparator-based interrupt when the capacitor voltage (V_C) falls below a lower threshold (V_{THR_L}). Subsequently, the MCU can wake up[4] again if another such interrupt is set at $V_C >$

[4]The plots in Figure 3.9 do not show the wakeup-time durations for clarity.

V_{THR_H} [152]. In this case, a *protocol step*, such as a MAC calculation, can have multiple *processing steps*.

Another way could be to go to sleep *after* each protocol step in order to reduce the number of wakeups and the associated delay at the cost of a larger capacitor. Here, the protocol step is the same as the processing step (Figure 3.9b). In this case, the supercap size should be chosen based on the most energy-consuming protocol step. However, this can be problematic if such a step is changed in the future due to the reprogramming of the IMD with a different authentication protocol. Note that in this scheme as well the comparator interrupt will be required to wake up the device, indicating that the capacitor has been sufficiently charged.

3.5.3.2 Gradual switch to harvested energy

In another approach, the implant can use the battery for the first authentication request and if it fails, it can switch to harvested energy for subsequent accesses within a specified time-frame. This can allow for smaller reservoir sizes since we can afford the resulting delay due to frequent charge/discharge cycles in case of an illegal entity.

3.5.4 Timeouts

It can be argued that timeouts can be employed as a simpler alternative to ZPD. For instance, after a certain number of incorrect attempts, the IMD can be made to not accept further messages for a certain duration. For domains other than IMDs this can be a natural choice. However, for IMDs, these timeouts can significantly compromise patient safety. For instance, any timeout after a malicious access can subsequently block a valid authentication attempt, which impacts *availability*.

3.5.5 Standalone ZPD module

As discussed in Section 5.5.4, the ZPD circuitry should not impact the already-constrained design choices from the manufacturer's perspective. When incorporating ZPD, it is likely that the manufacturer's preferable course of action would be to retain most of the existing IMD design in order to expedite regulatory approval. A solution to this problem is to design a ZPD module that sits externally to the main IMD core next to the antenna and is minimally invasive from the IMD design perspective. This is shown in Figure 3.10.

This module decouples the antenna from the rest of the IMD with the help of a switch. The antenna is initially disconnected from the IMD transceiver. Upon receiving the incoming RF, the ZPD module is powered up using the harvested energy and executes the authentication protocol. When the external entity/reader is authenticated, the ZPD module turns on the switch so that the IMD is able to communicate with the reader in a secure manner. Upon completion of the communication session, the ZPD module turns off the switch. This configuration, however, poses two new

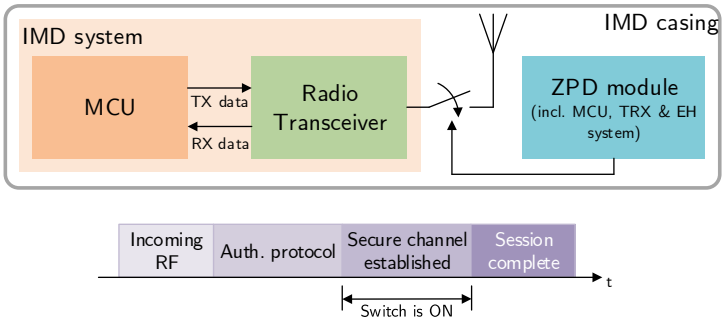


Figure 3.10: *Standalone ZPD module*

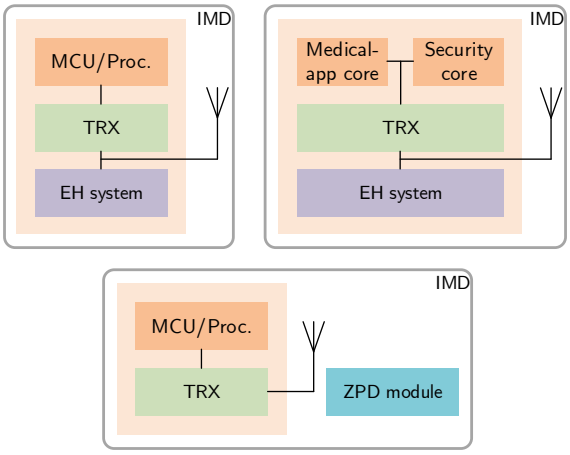


Figure 3.11: *Taxonomy of ZPD implementations. Top left: single-processor implementation. Top right: dual-processor implementation. Bottom: standalone ZPD module with the internal IMD design unchanged.*

constraints: (1) There should be enough space inside the casing for the placement of this standalone module. (2) The antenna (or coil) used by the ZPD module should not be obstructed by the metallic casing. Otherwise, it can negatively impact energy harvesting and wireless communication.

Regarding the first constraint, we observe that it is quite common for the IMDs (e.g., [102]) to have sufficient vacant space inside the metal casing. Regarding the second constraint as well, we have examples of rechargeable IMDs such as [122], which have unusually large charging coils compared to non-rechargeable IMDs. Here, the coil is embedded within an elastomeric plate, which is placed outside the IMD's titanium casing. This allows unobstructed WPT. Hence, it is reasonable to assume that the IMDs currently in the field can accommodate a standalone ZPD module.

Table 3.4: *Comparison of ZPD-enabled IMD designs with respect to a base, single-processor, non-ZPD system.*

	1P-ZPD	2P-ZPD	S-ZPD
Function-DoS protection	0	+	+
Design suitability			
Certification effort	−−	− − −	−
Area cost	−	−−	−−
Power/Energy cost	−	−	−
Maintainability	−	+	+
Reliability	0	0	0

(+) stands for a benefit, (−) for a drawback and (0) for no perceptible change compared to the base design.

As discussed in Section 3.2, the IMD transceiver usually polls for an external entity by cycling through sleep and sniff modes. Employing the above-discussed standalone ZPD module alleviates the need for such polling, and thus, the transceiver can completely stay asleep. However, this implies modifying the transceiver functionality, which was intended to be avoided in the first place. Thankfully, such changes can be performed at the firmware level, which are far less invasive than changing the transceiver circuitry.

3.5.6 Taxonomy of ZPD implementations

In terms of implementing ZPD, we can have three possible schemes, as shown in Figure 3.11.

1. Single-processor ZPD (1P-ZPD): This is the most basic implementation in which a dedicated energy-harvesting system is added to a reference single-processor IMD; see Figure 3.2. This processor is responsible for both executing the medical application and for receiving/sending data packets from/to the transceiver. This handling of data includes running the crypto primitives in order to authenticate the external device (using harvested energy).

2. Dual-processor ZPD (2P-ZPD): This scheme was originally proposed in [164], in which a dedicated processor is added (within the IMD system) for handling the communication data and executing the security primitives. This helps protect the IMD against function DoS.

3. Standalone ZPD (S-ZPD): This type of ZPD implementation is the scheme introduced in Section 3.5.5. Note that similar to 2P-ZPD, this scheme has a second processor as well, which is inside the standalone module.

These implementations are compared in Table 3.4 against a reference, non-ZPD design. Only 2P-ZPD and S-ZPD provide protection against *function DoS* since the

medical and security tasks are decoupled and executed on separate processors. In addition, S-ZPD provides the fastest time to market because of a significantly shorter approval cycle of the standalone module. On the other hand, as evident from Figure 3.11, 1P-ZPD results in the lowest area overheads compared to the other two schemes. Note, however, that 2P-ZPD and S-ZPD do not introduce significant energy and power costs since the authentication is performed by the additional processors using only harvested energy and, after authentication, these processors can enter their deepest sleep modes.

In terms of *maintainability*, both 2P-ZPD and S-ZPD decouple the security-related processing from the main implant functionality. This makes it straightforward to update the security firmware, without the need for touching the medical application. Hence, the potential maintainability cost of 1P-ZPD is considerably higher than the other two schemes.

In order to evaluate the schemes in terms of *reliability*, we consider *functional safety*, which is based on the industry-established meta-standard IEC 61508 [69] and has been increasingly used for a diverse number of application domains, ranging from cars, and planes to IMDs.

Functional safety can be calculated via such techniques as Failure Mode and Effect Analysis (FMEA). More specifically and without loss of generality, we consider here an IMD processor comprising three critical subparts: the core, the instruction memory (FLASH) and the data memory (SRAM). We, then, proceed to perform an FMEA on the *safety functions* included in the IMD safety-critical systems; in this case the aforementioned three subparts. The objective is to calculate the Probability of Failure per Hour (PFH), an absolute metric for the overall system. For constant failure rates, PFH is given by (3.8):

$$PFH = \sum \lambda_{DU} \tag{3.8}$$

Here, λ_{DU} is the rate of dangerous undetected failures observed. PFH encompasses a λ_{DU} per each of the sub-components (i.e., here, IMD subparts) of the analyzed safety function and accounts for both permanent and transient faults. Here, we draw the λ_{DU} values of all such components from confidential industrial data [92] in the possession of YogiTech S.p.A., now an Intel company. This dataset is empirically collected and pertains to an 18-nm process technology and, hence, is relevant for future IMDs, as well. Considering a *high-demand* application scenario – since we focus on IMDs –, we can use calculated PFH figures to also derive the well-known FIT metric (Failures In Time) based on well-known values (see Table 3.5).

Based on the above, the reliability findings for the three implementations are collected in Table 3.6. Since the S-ZPD module is standalone, it must contain its own processor (or MCU). Then, the amount of processing logic and memory footprint of this scheme is similar to that of 2P-ZPD, which is roughly double to that of 1P-ZPD. Hence, 1P-ZPD should have a lower PFH compared to the other two

Table 3.5: *High-demand PFH for an 18-nm technology node*

PFH	FITs
$< 10^{-5}$/h	10000
$< 10^{-6}$/h	1000
$< 10^{-7}$/h	100
$< 10^{-8}$/h	10

Table 3.6: *Reliability evaluation of ZPD-enabled IMD designs*

	PFH	FIT
1P-ZPD	4.86×10^{-8}	100
2P-ZPD & S-ZPD	9.70×10^{-8}	100

schemes (assuming same processor architecture and memories are used in all the schemes). We take as an example a 16-bit 5-stage RISC processor with a separate 16-kB instruction (FLASH) memory and a 16-kB data memory (SRAM). Incorporating an additional processor (i.e., in schemes 2P-ZPD and S-ZPD) doubles the PFH value to that of 1P-ZPD. However, there is no change in FIT value, due to the trivial silicon overhead involved in moving from a single to two tiny implant processors due to the lightweight processor designs used [164]. Hence, 2P-ZPD and S-ZPD do not impact the IMD reliability perceptibly.

3.6 Summary

Over the last few years, energy harvesting has been presented as the most effective solution for protecting IMDs against battery-depletion attacks. In this chapter, we have provided an extensive review of IMD-specific ZPD works from literature. We analyzed these works based on our formulated design considerations, and highlighted their shortcomings. This work is the first to substantiate these considerations and to provide recommendations towards practical ZPD implementations. These include, among others, the concept of adaptive ZPD with the purpose of facilitating bedside-base-station operation, and the standalone ZPD module with the aim of reducing the IMD certification effort and the time to market.

CHAPTER 4

"Whenever I made a roast, I always started off by cutting off the ends, just like my grandmother did. Someone once asked me why I did it, and I realized I had no idea. It had never occurred to me to wonder. It was just the way it was done. Eventually I asked my grandmother, *"Why do you always cut off the ends of a roast?"* She answered, *"Because my pan is small and otherwise the roasts would not fit.""*

David Basin (related this allegory to highlight current practices in security-protocol design), PROOFS 2018 keynote

Architecting a secure protocol for implantable medical devices

M. A. Siddiqi, C. Doerr, and C. Strydis, "IMDfence: Architecting a Secure Protocol for Implantable Medical Devices," *IEEE Access*, vol. 8, pp. 147 948–147 964, 2020.

M. A. Siddiqi and C. Strydis, "IMD Security vs. Energy: Are we tilting at windmills?: POSTER," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*. ACM, 2019, pp. 283–285.

Earlier-generation IMDs had little or no security provisions whatsoever, as confirmed by numerous ethical-hacking incidents over the past decade [62, 94, 95]. The research community has responded with a wealth of new schemes and, eventually, top IMD manufacturers now claim to have rectified the security weaknesses over the past few years [50, 105].

However, due to the constraints imposed by an IMD's scant computational, storage and energy resources, most proposed schemes in research have refrained from taking proven security approaches. Moreover, since these schemes have been specifically tailored for IMDs, they have missed the big picture and resulted in limited coverage of the security properties essential to a modern IMD. Specifically, focus has mostly been drawn on confidentiality, integrity, authentication and emergency access (e.g., [15, 16, 23, 29] etc.), while non-repudiation, remote monitoring and system scalability have been left unaddressed for the most part. Besides being difficult to tackle, prior seminal work has not identified or stressed the importance of these additional requirements.

In this chapter, we debunk the myth that advanced security is impossible in modern IMDs. To this end, we collect both well-studied and overlooked security requirements, impose strict design constraints, and propose IMDfence, a novel security protocol for IMD ecosystems. This chapter contributes:

- A comprehensive security protocol for a modern IMD ecosystem, IMDfence, which addresses crucial, yet previously ignored requirements, i.e., non-repudiation, remote monitoring and system scalability.
- A realistic solution for accessing the IMD during emergencies without compromising security or patient safety.
- A rigorous evaluation of IMDfence paying special attention to the protection against battery denial-of-service (DoS) attacks.

The rest of the chapter is organized as follows: We enumerate modern IMD-system requirements in Section 4.1, and then discuss existing systems in Section 4.2. Section 4.3 details our proposed security protocol. We evaluate IMDfence in Section 4.4. Section 4.5 reviews the related work. We provide concluding remarks in Section 4.6.

4.1 IMD-security requirements

In this section, we collect and present the necessary security and related functional requirements (SRs) that should be satisfied in modern IMD systems. These requirements, which were briefly discussed in Chapter 2 (Section 2.2.3), form the basis of the IMD-specific security protocol, to be detailed in Section 4.3.

4.1.1 Basic security services (SR1 & SR2)

As in other domains, the IMD-security system should provide the fundamental security services: *Confidentiality*, *Integrity* and *Availability*. The first two services (SR1) are usually addressed through the use of lightweight block-ciphers and message-authentication codes (MAC) [164]. More specifically, the commands sent from the reader to the IMD and the associated responses (e.g., data logs) should be treated as confidential and it should be ensured that such data is not modified in transit.

Availability ensures that the IMD is always available for patient treatment whenever required (SR2). This implies that the device should be protected against Denial-of-Service (DoS) attacks. One of the highest-likelihood and lowest-cost attacks is the battery-depletion attack (or battery DoS attack), as indicated in Chapter 2 and discussed in detail in Chapter 3.

4.1.2 Non-repudiation (SR3)

Non-repudiation ensures that the sender of a message is not able to deny (or *repudiate*) its creation. Since there is always a possibility of malpractices, medical mistakes or insider attacks, we require non-repudiation to aid in computer forensics in case a patient experiences medical issues as a direct consequence of such actions. This security service ensures that a physician, paramedic or nurse is not able to deny his/her involvement in such scenarios. Non-repudiation has not been given due consideration by the research community when it comes to IMD systems. One of the reasons is that *true* non-repudiation can only be achieved through the use of public-key (or asymmetric) cryptography for computing digital signatures [177], which has traditionally been considered to be too resource-costly for IMDs [148,164]. Another, very important, reason is that past generations of IMDs could only be accessed by one person, i.e., the physician. Nowadays, the IMDs can be accessed by multiple people, including the patients themselves [103, 160, 162]. Hence, there is a need to introduce *user accountability*.

Most of the existing IMD-security works have looked into strict reader-IMD communication (without the involvement of a trusted third party). Even if we assume that the resource-constrained IMD is able to support public-key computations, this reader-IMD configuration makes it impossible for the IMD to effectively use public-key cryptography since it cannot keep track of the validity of the reader certificates (due to lack of Internet connectivity). What is more, these devices do not have sufficient memory to store the required certificates [96]. For instance, the IMD must store all possible reader certificates if we want to support access during travels or when the patient is visiting abroad. Hence, a scheme is required that employs *additional* architectural components (as will be discussed in Section 4.3) to solve these issues.

Another complication is the legal aspect. Since non-repudiation is there to provide evidence, it should be incorporated based on the assumption that such ev-

idence will be scrutinized by a hostile legal expert [7]. One main limitation of cryptography-based non-repudiation is that there is no formally-verifiable link between the device that signs the digital signature and its user. For example, the user, i.e., the private-key owner, can falsely claim that the signature has been generated by a malware program without his/her consent, or that the private key has been stolen. There is no technical mechanism that can determine whether such a claim is false [128]. The IMD security protocol should address this limitation, which we term as the *Non-repudiation gap*.

4.1.3 Emergency access (SR4)

Patient safety always outweighs device security. Hence, during emergencies the security protocol should not hinder or delay paramedic access to the IMD [131, 139]. Although it seems reasonable to drop security altogether in such situations, this can be a problem if, while in a normal mode, an adversary fools the IMD into entering the emergency-access mode. The security protocol must be capable of allowing the IMD to accurately *classify* whether a communication attempt is an emergency or a normal access. This ensures that the adversary is unable to trigger and exploit the emergency-access mode. Furthermore, since there is a high likelihood of the patient losing control of his/her actions in emergencies, the emergency-access mode should be independent of patient participation.

4.1.4 Multi-manufacturer environment (SR5)

Past works on emergency access have ignored the fact that, in emergencies, it is unlikely for the paramedic to know the IMD make and model beforehand. Moreover, it is not possible to preemptively stock all the readers from all the manufacturers in the ambulance. Hence, to achieve *true* emergency access, the IMD-security system should be *manufacturer-independent*, i.e., all manufacturers need to agree on a unified standard for secure reader-IMD communication. This way, an ambulance can use one generic reader regardless of the IMD manufacturer and type. It follows that an emergency-access scheme should be adoptable by all IMD types. E.g., an emergency-access solution that requires an IMD measuring the cardiac signal [131], can be easily incorporated in pacemakers, but it will require significant modifications in neurostimulators.

As things stand, true emergency access does not exist in commercial IMDs. As long as this remains acceptable to the medical community, SR5 can be relaxed. This is further discussed in Section 4.3.4.2.

4.1.5 Access control (SR6)

The access privileges of the reader should be differentiated based on the type of user. For example, nurses, patients or patient relatives may only be allowed to read status data from the implant, whereas a physician and a paramedic may further be

allowed to modify the implant configuration for therapy updates, suspend or resume its operation. Similarly, a technician may be allowed to modify the implant firmware in addition to tasks of the above user roles.

4.1.6 User and reader-IMD authentication (SR7)

In order to aid in non-repudiation and access control, the IMD system should be able to identify the physician, nurse, paramedic etc. who is using the reader to communicate with the implant. Similarly, the reader should also be able to authenticate the IMD in order to prevent spoofing attacks on the reader. Hence, there is a requirement for performing *mutual* authentication instead of just authenticating the reader unilaterally [164]. Furthermore, said authentication is required to be *strong*, i.e., it should imply both *message* and *entity* authentication, and guarantee message *freshness*, or in other words *replay protection.*

4.1.7 Flexibility and scalability (SR8)

The IMD should not be limited to communicating with only a fixed amount of readers since this severely limits portability, e.g., during emergencies when a paramedic reader is used, or when there is a need for treatment at some hospital during travels. Hence, there should not be any pre-shared secrets between the reader and IMD.

4.1.8 Bedside-reader operation for remote monitoring (SR9)

Some of the modern IMD systems also include a bedside reader, which enables remote monitoring [159]. It establishes communication with the IMD when the patient is asleep and sends treatment status to a back-end server via an Internet connection. However, this additional connection represents an increase in the attack surface, which imposes additional security requirements. We predict that the use of such readers will become more widespread over time due to their time- and cost-saving features. Hence, this phenomenon should proactively be considered when designing secure IMD systems.

4.2 Existing systems

IMD manufacturers have typically relied on “security through obscurity”, i.e., they choose to hide the communication-protocol specifications in order to enhance security. This is not a recommended practice, and as a consequence of using this approach, we have seen several successful blackbox-hacking attempts over the past few years [62, 94].

Some of the latest commercial IMDs, including neurostimulators [162], insertable cardiac monitors [160] and even pacemakers [103] offer a Bluetooth Low Energy (BLE) connection between the patient smart-phone and the implant. The initial pairing between these devices is based on the BLE standard in addition to proprietary

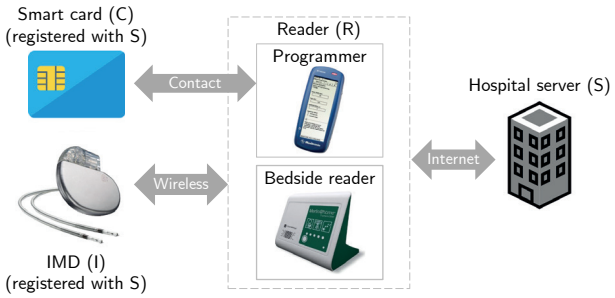


Figure 4.1: *Proposed IMD ecosystem*

protocols [162]. However, they do not disclose the *association models* used in these pairings, which makes these devices vulnerable to attacks due to the reasons mentioned above. In most of the cardiac devices, in the absence of an IMD-programmer, a magnet can be used to disable therapy or to switch to a default behavior [161]. This mode, however, can be easily exploited by adversaries through the use of a strong magnet when in close proximity to the patient (e.g., in public transport).

4.3 IMDfence: Security protocol for IMD ecosystems

The absence of a complete security solution for IMD systems has led us to propose IMDfence, a novel secure-communication protocol that satisfies the extensive and strict requirements enumerated in Section 4.1. As will be shown, IMDfence addresses the complete IMD ecosystem.

4.3.1 Configuration and assumptions

The IMDfence configuration includes a smart card (C) for the user (U) trying to access the IMD (e.g., a physician), and a trusted third party (TTP), i.e., a hospital server (S), in addition to the implant (I) and the reader (R); see Figure 4.1. The list of notations used in this chapter is summarized in Table 2.1. The extra components, C and S, are employed to facilitate ***non-repudiation (SR3), access control (SR6)*** and ***user authentication (SR7)***, as identified in Section 4.1. Each personal smart card, which is inserted in R, supports public-key cryptography. Its private key, which is unique to each card/user, enables digital-signature computation, thus providing non-repudiation. Since R and C are untrusted with respect to each other, a TTP (S) is required to mutually authenticate the two entities. Non-repudiation can technically also be provided through the use of a *personal* reader that supports public-key computations in order to get rid of C and S. However, such a solution would be highly impractical and expensive since it would require all the doctors and nurses to be in possession of their personal readers at all times. Moreover, the use of S also enables access control and facilitates ***bedside-reader operation (SR9)***.

Every user requires their own C and should know the associated PIN (two-factor authentication). Since patients are only allowed *read-only* access (as discussed in Section 4.1.5), losing or misplacing their C will not inhibit any future treatment. To avoid additional attack vectors, we propose to not support the use of *contactless* smart cards and magnetic-strip cards.

4.3.1.1 Interfaces

For tackling ***flexibility and scalability (SR8)***, there is no pre-shared key between $R \leftrightarrow I$, $R \leftrightarrow C$, $S \leftrightarrow R$, and $C \leftrightarrow I$. The only pre-shared symmetric keys that exist are between $S \leftrightarrow I$ (K_{SI}) and $S \leftrightarrow C$ (K_{SC}). A unique K_{SI} is installed in the implant at the time of manufacturing, which is then shared with the server of the hospital where the implantation surgery is going to take place. During this IMD-registration process, the implant is also assigned a unique and random identifier ID_I, which is stored in the implant. Likewise, K_{SC} is installed in the smart card and is shared with the hospital where the card user is registered. Moreover, S, I and C can only talk to R directly and only indirectly with each other[1].

The secure communication between $S \leftrightarrow R$ is made possible by employing public-key-based key exchange in which the public/private key pairs of these entities are used. This configuration helps in making R independent of the need to pre-share keys with the hospital, which aids in scalability. As a result, a patient can use his/her personal reader from any location, and/or buy a new reader from the manufacturer without the need of registering it first at the hospital.

In our proposed configuration, each smart card also has its own public/private key pair. Technically, R has the capability of maintaining a comprehensive certificate-revocation list (CRL) of smart cards due to frequent Internet connectivity. Hence, it is able to verify smart-card certificates. On the other hand, due to the limited on-board memory and less-frequent Internet connectivity, C can only maintain a small CRL that does not change frequently. Hence, C can not verify the authenticity of the multitude of reader certificates. As a result, public-key-based key exchange cannot be used to establish a session key between $R \leftrightarrow C$. However, it will be shown in Section 4.3.3 that the session key between $R \leftrightarrow C$ will be established using S as a TTP. The same will be done for establishing a session key between $R \leftrightarrow I$. Lastly, no session key is required between $C \leftrightarrow I$.

4.3.1.2 Centralization and Public-key infrastructure

The public keys of S, R and C are signed by a trusted certification authority (CA) belonging to the manufacturer. The smart-card certificates, in addition, also include the user privileges.

We consider the precise implementation details of public-key infrastructure (PKI) and certificate revocation outside the scope of this chapter. In case of a smart card,

[1] The routing details of the messages communicated via the reader have been omitted for brevity.

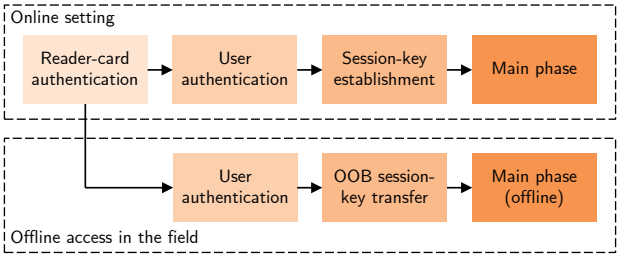


Figure 4.2: *IMDfence flow under online and offline scenarios*

certificate revocation would be needed when a card is stolen, a user leaves, or he/she changes roles (e.g., from nurse to paramedic). For a reader, certificate revocation would be required in case R is stolen or deemed as out-of-service. The server is given the responsibility to verify the certificates of R and C and hence, it is assumed that it maintains an up-to-date CRL.

4.3.1.3 Modes of operation

We propose two modes of operating in IMDfence, one for regular (online) operation and the other in the absence of an active Internet connection (offline), e.g., during ***emergencies (SR4)***; see Figure 4.2. Online mode offers the full security- and functional-requirement portfolio highlighted in Section 4.1, whereas offline mode results in the graceful degradation of offered services without compromising security and patient safety. Since S is not available in offline mode, R and I will be required to undergo an out-of-band (OOB) pairing phase in order to securely exchange a short-term session key. These modes and the constituent phases will be elaborated in the following sections.

4.3.2 Threat model

Building on the attacker model described in Chapter 2 (Section 2.2.2), R is assumed to be untrustworthy by I, C and S, and vice versa. Moreover, we assume that if A steals a personal smart card or a valid reader, then the user or hospital staff should notify the hospital server so that it is blacklisted. Additionally, we assume that A can hack the reader to read out or modify data at the interface of the inserted smart card. However, A does not have access to the keys stored in R and C. This implies that protection against side-channel attacks is considered outside the scope of this work since such attacks are typically addressed through specialized countermeasures. Moreover, due to the assumption that S is notified of a lost/stolen device, A has a limited time window to perform such attacks after stealing a device. We also assume that the hospital personnel do not have access to the keys stored in the server since such attacks can be prevented by employing standard practices, such as hardware security modules (HSM) etc.

4.3.3 Regular (online) mode

The regular mode of IMDfence is shown in Figures 4.3 to 4.6. It starts with the $R \leftrightarrow C$ *mutual authentication* phase after the physician (or any other user) inserts their smart card into the reader.

4.3.3.1 $R \leftrightarrow C$ mutual authentication

In this phase, R first tries to establish a secure connection with S by sending its identifier and a nonce. In order to deter distributed-denial-of-service (DDoS) attacks against S (to ensure server ***availability (SR2)***), a basic client-puzzle protocol (CPP) is employed [73]. CPP is a proof-of-work system in which any client (or in this case a reader) that wants to access the server (during high load) is required to correctly solve a cryptographic puzzle. For a single client the costs of solving this puzzle are negligible. However, in order to launch a successful DDoS by initiating a large number of simultaneous connections, it would be computationally infeasible for the attacker to solve a multitude of such puzzles.

S initiates CPP *if* it senses a DDoS attack or it is dealing with an abnormally high number of simultaneous connections. It first calculates x, which is the n-bit hash of ID_R, the current time stamp t and its long-term secret K_S. It then computes a second hash $\mathsf{h}(x)$. S sends $\mathsf{h}(x)$ and x *excluding* the first k bits of x, along with the t. R computes the solution, i.e., the missing k bits of x, and sends it along with ID_R and the received time stamp. k represents the difficulty of solving the puzzle. S calculates x again and verifies that the solution indeed corresponds to the missing bits. It also verifies, with the help of t, that the puzzle has not expired. S is protected against memory exhaustion since it is not required to store any data for the verification of the puzzle solution. In case these checks are successful, S sends its nonce to R.

R then performs a Diffie–Hellman (DH)-based handshake with S in which a session key is established between them based on their public/private key pairs (see Figure 4.3). During this handshake, both verify each other's certificates and, additionally, S checks if R is valid (i.e., it is not reported as stolen or out-of-service).

In order to achieve authentication between R and C, R then initiates a five-pass, mutual-authentication protocol borrowed from the ISO/IEC 9798-2 standard [71] with S acting as a TTP (see Figure 4.3). R and C ensure message freshness by exchanging their nonces in the first messages between them, and then verifying the existence of these nonces in the subsequent messages. R generates its nonce and sends it along with its identifier and N_S to C. C responds by generating N_C and sending a cryptogram (m_{SC_1}) that includes authenticated encryption of its certificate, ID_R and nonces, along with ID_C and N_C in plaintext. This cryptogram is calculated using K_{SC} since it is intended for the server. R stores ID_C and N_C, and forwards the cryptogram to the server, which establishes that it originated from C and that it is also tied to R. The server then verifies $Cert_C$ and checks the validity

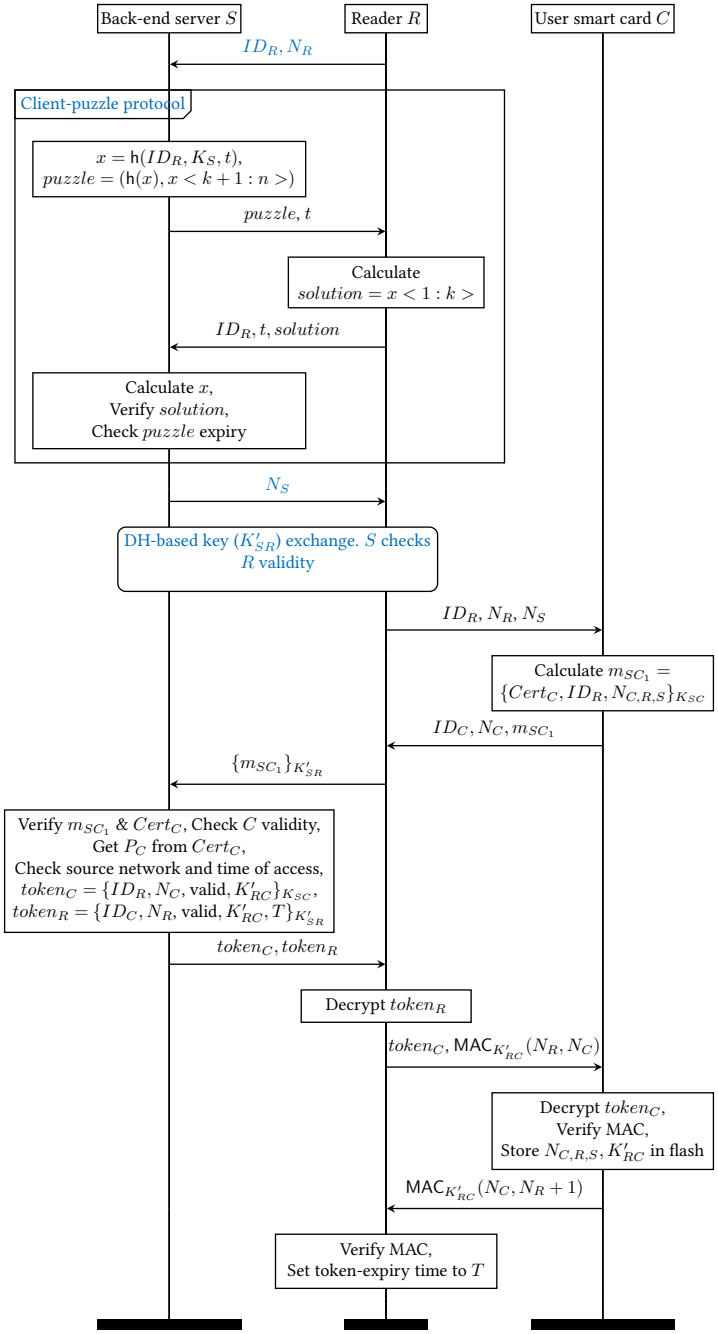


Figure 4.3: *Reader-card authentication. Steps that are common with bedside-reader mode are marked in blue.*

of C, in case it has been reported stolen or has expired. It then determines the required privileges (P_C) for the particular user (e.g., physician, paramedic, nurse etc) from $Cert_C$. It also calculates tokens for both these entities using the respective symmetric keys. These tokens include the nonces and identifiers of R and C and a fresh symmetric key K'_{RC}. Additionally, $token_R$ also contains T (reader-card-authentication lifetime). Based on these tokens, R and C can ascertain each other's trustworthiness.

R decrypts $token_R$, retrieves K'_{RC}, calculates the MAC of the nonces, and forwards it along with $token_C$ to C. The smart card similarly decrypts $token_C$ and verifies the received MAC using K'_{RC}. It stores the nonces and K'_{RC} in its internal flash memory[2] so that it can verify and create messages in the subsequent stages. C then sends a MAC that is calculated over N_R and N_C (including an addition by 1 to protect against replay of the previous message). R verifies the received MAC using K'_{RC}. At this point, both R and C have mutually authenticated each other.

R then sets its internal real-time clock to T and starts it to track the period over which the subsequent phases can execute without the need of reader-card authentication. Since it is possible that R is not connected to the Internet *during* its operation (e.g., in emergencies), this scheme enforces that R, by design, shall only be usable for a certain duration until it has first established an Internet connection. This makes sure that R receives critical firmware updates in time, if there are any. The selection and configuration of T will be discussed in Section 4.4.1.4.

4.3.3.2 User authentication

This phase is shown in Figure 4.4 and its objective is to authenticate the card holder. The physician enters his/her PIN using a keypad on the reader. R then checks its internal real-time clock to verify the validity of its token. R encrypts the PIN and the nonces (in order to prevent replays) using K'_{RC}. C decrypts the message using the same key, verifies the PIN by comparing it with the stored one and sends back a cryptogram intended for the server, which is encrypted with K_{SC}. It contains the confirmation of success in addition to the nonces.

4.3.3.3 Session-key (K'_{RI}) establishment

R then initiates a TTP-based key established protocol with S and I in order to acquire a symmetric session key K'_{RI} for providing ***confidentiality and integrity (SR1)***, as shown in Figure 4.5. R first exchanges the nonces and identifiers with I and then sends the nonces and identifiers of all parties to S along with m_{SC_2}. S first verifies m_{SC_2}. It then generates K'_{RI}, encrypts it in two independent messages m_R and m_I intended for R and I respectively, and then sends these to R. R decrypts

[2]There can be a time gap between this and the next stage (in offline mode). Since smart cards can only be powered by R, the above data has to be stored in the non-volatile (flash) memory so that C can be taken out of R during this period.

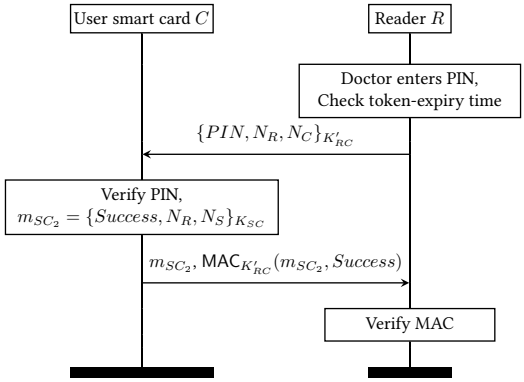


Figure 4.4: *User authentication at the reader*

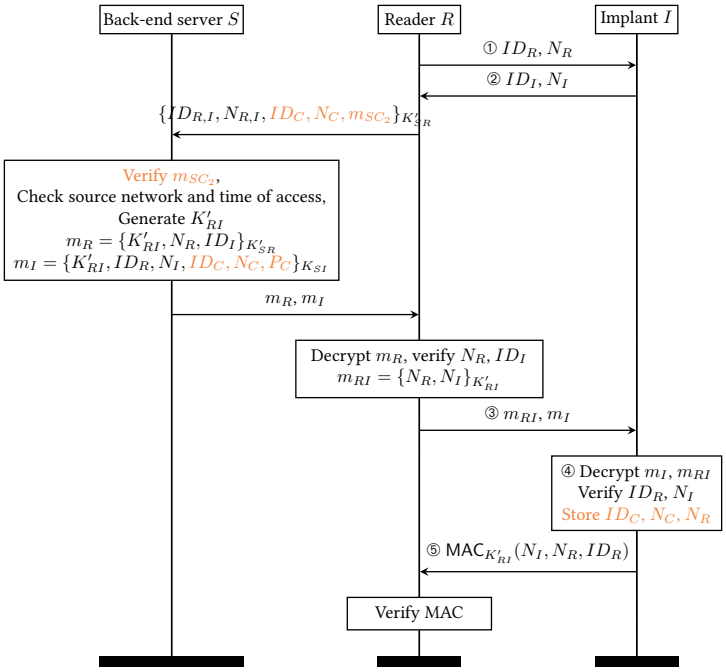


Figure 4.5: *Session-key establishment between R and I via S. Operations that are not relevant to bedside-reader mode are marked in orange.*

m_R and verifies its contents. It then encrypts N_R and N_I using K'_{RI} (to form m_{RI}) and then sends it along with m_I to I. I first retrieves K'_{RI} by decrypting m_I, and then decrypts m_{RI} to verify that R has the knowledge of K'_{RI} and that the nonces are valid. I finally creates a MAC using the new session key for R to validate. At the end of this protocol, both R and I are mutually ***authenticated (SR7)*** and have

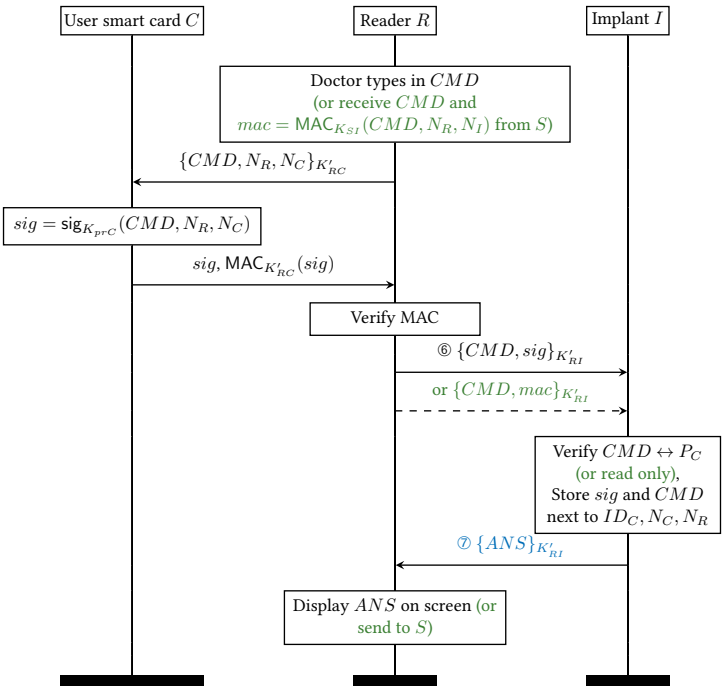


Figure 4.6: *Main phase. Steps that are common with bedside-reader mode are marked in blue. Operations that are unique to bedside-reader mode are marked in green.*

arrived at a *fresh* session key in addition to performing key confirmation. Similar to the reader-card authentication stage, this phase is also based on the five-pass protocol from ISO/IEC 9798-2 since it involves a TTP.

To protect against battery-DoS attacks (which impact ***availability (SR2)***), steps 1 to 4 of session-key establishment should be as lightweight as possible so that the IMD is able to execute it using harvested RF energy. This will be further discussed in Section 4.4.2.

4.3.3.4 Main phase

After session-key establishment, R allows the user to enter a command on the reader interface (see Figure 4.6). The command is encrypted along with the nonces (to prevent replay attacks) using K'_{RC} and is sent to C. The card decrypts the command, digitally signs the message using K_{prC} (to form sig) and sends it to R. R re-encrypts the command using K'_{RI} and sends it to the implant along with sig.

I decrypts the command and verifies if it corresponds to the privileges information received in m_I during the previous phase, hence ensuring ***access control (SR6)***. sig and CMD are stored by the IMD next to ID_C, N_C and N_R, which were stored during session-key establishment. This is required to ensure non-repudiation since

sig was signed using a personal private key. For example, in the case of a medical mistake (e.g., an incorrect command) that led to patient death, the physician will not be able to deny his/her involvement since this signature can always be retrieved from the IMD and subsequently verified using the associated data. It follows that signature storage is not required for *read-only* commands. Since the implant trusts the reader at this point, there is no need for I to verify the signature since the associated MAC has already been verified by R. This relieves I of the need to employ public-key cryptography and to track user certificates. After processing the command, the implant responds with an *answer* message encrypted with K'_{RI}. R displays it on its screen for the convenience of the user. The session keys expire after a *finish* command and its associated response, or after a period T.

4.3.3.5 Addressing the non-repudiation gap

As discussed in Section 4.1, the use of a signature alone is not sufficient to address the legal aspects of non-repudiation. In order to bridge the non-repudiation gap, one option could be to enforce that the user protects C and the associated PIN, or immediately reports in case it is lost. However, due to the possibility of human error in general, this is too much of a legal responsibility for the user.

A realistic way of bridging this gap is by introducing additional checks in the implementation of reader-card-authentication and session-key-establishment phases (see Figures 4.3 and 4.5, respectively). The server can ensure that the implant *write* access (determined from P_C) is requested from within the hospital network *and* during the working hours of the user. On the other hand, the server can allow *read-only* accesses from external networks, e.g., in case the access is made by the patient or their bedside reader. The user just has to ensure that R is issued from a certified repository, *and* that R should only be connected to a trusted Ethernet/Wi-Fi network (i.e., in a hospital or patient home). With these precautions, which a responsible user can easily follow, protection can be ensured against the malicious replacement of a command using a compromised reader, or against an attacker sending a malicious command him/herself in order to frame said user. Due to the above risk-based, multi-factor authentication, a user cannot falsely deny his/her involvement in a certain implant access because the alternative explanation implies that (1) the attacker stole a valid reader, card and pin, (2) accessed the implant from within the hospital and during the user's working hours, and (3) R and C were not reported as stolen. The combined probabilities of all these events occurring at once is extremely small, or, in other words, the non-repudiation gap is effectively bridged by the introduction of above checks.

4.3.3.6 Bedside-reader operation

The online mode also facilitates bedside-reader operation (see Figure 4.1). Here, only the CPP and DH-based handshake between the bedside R and S (from reader-

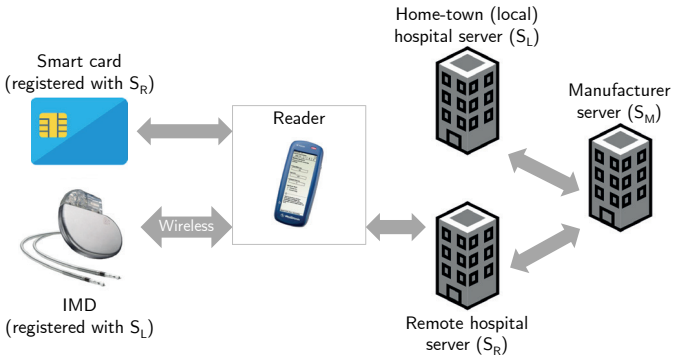


Figure 4.7: *Scenario when the patient is out of town*

card authentication phase), the session-key establishment phase, and the main phase (with a few differences, as indicated in Figure 4.5 and Figure 4.6, respectively) need to be executed, since the commands and responses are only sent and read by S. Moreover, since the remote monitoring done in practice is only *read only*, i.e., with the lowest access privileges, there is no need for non-repudiation *if* the read-only access control is implemented correctly. This can be done if sig in step 6 is replaced by MAC of CMD from S (i.e., $\mathsf{MAC}_{K_{SI}}(CMD, N_R, N_I)$). Using this MAC, I is able to verify that the command came from the server, and hence, it can be executed with read-only privileges. Finally, the hospital staff can retrieve the critical treatment data by logging into S. It can be argued that this remote-access mode should support *read/write* access instead of just read-only in order to enable remote firmware updates. However, we stress that such updates should always occur in the presence of a qualified professional. This is important in case patient health suddenly deteriorates due to the update process. Moreover, in practice it is quite common and acceptable to get the IMD firmwares updated at the clinic in the presence of a physician [50]. This mode is also useful for securely retrieving the stored signatures pertaining to previous programming sessions in order to free up limited IMD memory.

4.3.3.7 IMD access from a non-local location

In Section 4.3.1.1, we discussed that C and I are registered at the local hospital (S_L), or in other words, they share their respective symmetric keys with the hospital server. During travels or when the patient is out of town, a situation may arise that requires access to the IMD for status monitoring. In this case, the scheme from Figure 4.1, can still work if the patient is in possession of R and his/her C. However, for treatment updates, which require higher access privileges, the patient would need to visit a nearby (remote) hospital (S_R). In this case, the above scheme would not work straightaway since the IMD is not registered at S_R and the remote-

location physician's C is not registered at S_L. Hence, minor extensions are required (see Figure 4.7), in which S_R establishes a secure connection with S_L via an IMD-manufacturer server S_M. S_M maintains a list of all the IMDs in service and the hospitals at which they are registered. Based on ID_I sent by R to S_R (and then S_R to S_M) during the session-key establishment phase (see Figure 4.5), S_M determines S_L and establishes a secure connection with it. S_R sends K'_{RI}, the relevant identifiers, nonces and P_C to S_L (via S_M) so that S_L is able to construct m_I and send it back to S_R. The protocol then proceeds normally and the IMD eventually retrieves K'_{RI} after decrypting m_I.

4.3.4 Offline mode

In the absence of an active Internet connection and hence, the TTP (S), e.g., during emergencies, R and I need to establish a *temporary* shared key so that they can communicate directly in a secure manner. We propose to employ an OOB-channel-based key exchange while using the principle of *touch-to-access*, which was discussed in Chapter 2 (Section 2.2.5.1). This principle is employed by I to establish trust with R since we assume R to be untrustworthy from the perspective of the IMD. We propose to either use *ultrasound communication* or *galvanic coupling* as the OOB channel (between R and I) since they result in virtually zero information leakage compared to other coupling methods, such as capacitive coupling [173]. Moreover, they have an advantage over biometric-based touch-to-access mechanisms (which will be discussed in Section 4.5) in that they do not require any initial RF communication messages before the IMD is sure that the external entity is in close proximity. This provides an additional security layer, which is critical for the pre-deployment configuration that will be discussed in Section 4.3.4.1. In Chapter 5, we will revisit the issue of OOB-based device pairing and investigate the applicability of ultrasound waves for it.

The paramedic places the OOB interface of the reader on the patient skin[3] at a point that is nearest to the IMD. The patient is assumed to thwart advances of a stranger trying to place a reader on his/her skin, if there is no emergency or a need for treatment. Hence, the implant assumes that the message received from the OOB interface is from a trustworthy source. In other words, in offline mode, the IMD-system security hinges on this OOB pairing and favors availability over security but in a more controlled fashion than state of the art.

The protocol is shown in Figure 4.8. The paramedic is required to perform reader-card authentication when starting his/her duty, so that both R and C obtain their respective tokens from S. When IMD access is required in an offline setting, R first initiates user authentication with the paramedic smart card in the same way as in the regular mode. During user authentication, R verifies that its internal

[3]Touching the skin is mandatory for the galvanic channel to function. Same applies to MHz-range ultrasound communication, as will be discussed in Chapter 5.

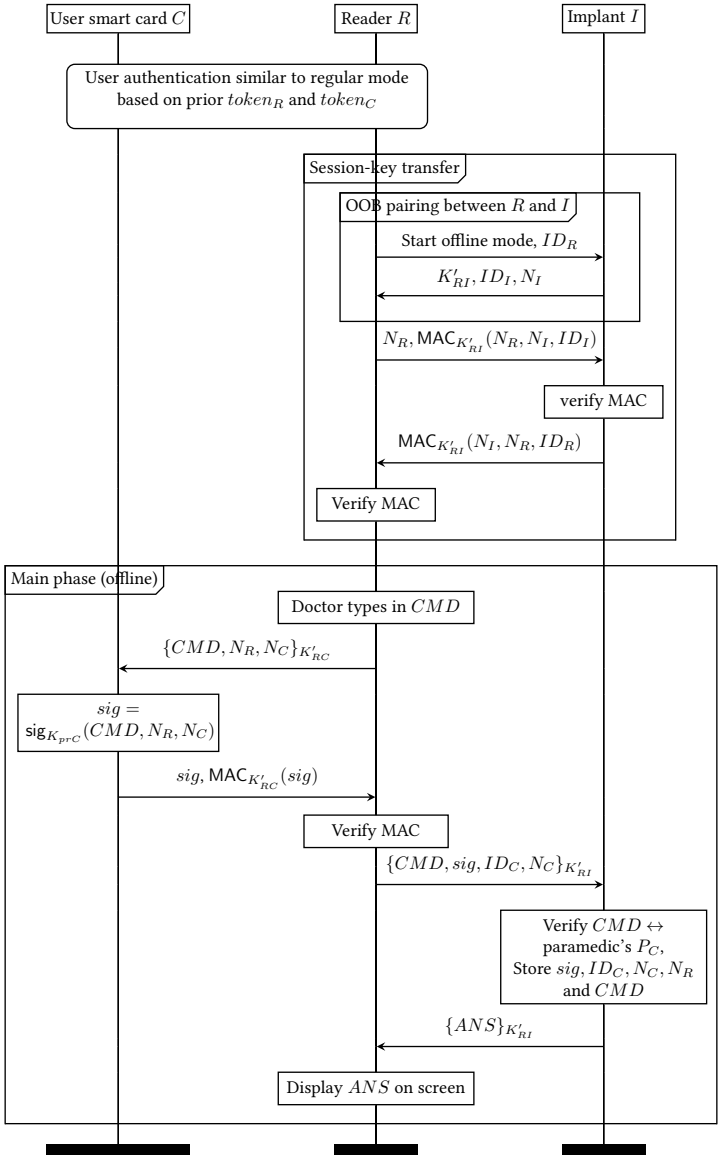


Figure 4.8: *IMDfence (Offline mode)*

real-time-clock value is less than T. Through the OOB channel, R sends a request for offline access along with its identifier. Upon receiving this request, the implant assumes that this is an offline scenario since this channel is activated only in such extraordinary circumstances. As a result, it generates a random key K'_{RI} and its nonce and sends them along with ID_I to the reader using the same channel.

R, then, initiates session-key confirmation with I in which both entities verify each other's MACs that are generated using K'_{RI}. In order to update or inquire about the implant operation, the paramedic enters the command on the reader interface, which is encrypted using K'_{RC} and is sent to C. The card digitally signs this command and sends it back to R. R encrypts the command using K'_{RI} and sends it to the implant along with sig, ID_C and N_C. This signature and CMD are stored by the IMD and are required to ensure non-repudiation, as already discussed in Section 4.3.3. The IMD responds with an answer encrypted by the same session key, which is subsequently displayed on the reader display. The session key expires in a manner similar to that in the regular mode.

In offline mode, the user is only allowed paramedic-level privileges, which have less access rights compared to a technician (see Section 4.1). The use of the OOB channel makes it straightforward for the IMD to decide on granting only paramedic-role commands.

4.3.4.1 Offline access with/without non-repudiation and access control

We also propose a second flavor of the offline mode in which non-repudiation and user authentication are not a requirement. This is suitable for less critical implants, such as neurostimulators. This flavor does not require a smart card, and as a result we do not require the reader-card- and user-authentication phases in addition to signature generation. This improves *usability*, since the paramedic is not required to perform reader-card authentication when starting their duty. In this scheme, the touch-to-access principle is deemed to be sufficient in order to ensure trust establishment. It is important to note that, for IMDfence, supporting non-repudiation during offline mode has to be decided *before* IMD-system deployment since it cannot be configured at runtime, so as to avoid exploitation.

4.3.4.2 Offline access with/without reader-interface standardization

As indicated in Section 4.1, supporting emergency access in the field requires a standardized reader interface, which demands collaboration between major IMD manufacturers. In order to facilitate this ***multi-manufacturer environment (SR5)***, there has to be one agreed-upon root CA that grants certificates to the manufacturers, who can then act as intermediate CAs that sign public keys of S, R and C. As things stand, however, *true* emergency access does not exist in commercial IMDs. As long as this remains an open issue, the above standardization is not required, and as a result, IMDfence can be simplified by eliminating the need for a global root CA. Emergency-access support in IMDfence is intended to be there in anticipation of any future changes in this regard.

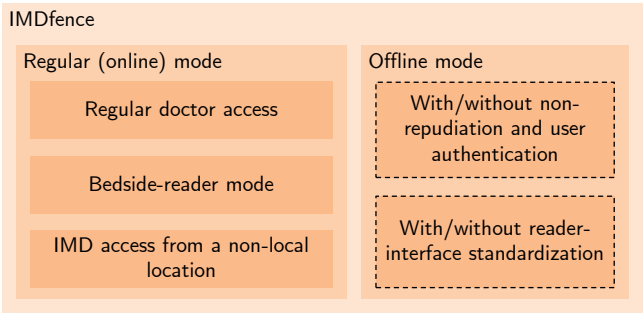


Figure 4.9: *IMDfence configurations and use cases*

4.3.5 Summary of protocol configurations

The different configurations of IMDfence are highlighted in Figure 4.9. The dotted boxes indicate (fixed) pre-deployment configurations, which cannot be changed at run-time. Such configurations were discussed in Sections 4.3.4.1 and 4.3.4.2.

IMDfence is designed in such a way that an attacker cannot target one mode over another for exploitation. For instance, the offline mode is only triggered after an OOB access, which is protected by the touch-to-access principle. Moreover, the sub-modes of online access only come about by disabling certain IMDfence steps instead of switching to a totally independent behavior.

4.4 Evaluation

In this section, we evaluate our system in terms of security feasibility and also look into the handling of battery-DoS protection for IMDs.

4.4.1 Security analysis

4.4.1.1 Automatic validation using AVISPA tool

For the automated and formal validation of IMDfence, several tools were available: AVISPA [170], ProVerif [19], Scyther [35] and Tamarin [107], among others. We eventually selected AVISPA (Automated Validation of Internet Security Protocols and Applications) since it is sufficiently capable of validating the handshake-specific protocol requirements based on its extensive use for similar protocols, while at the same time being simple to use. Moreover, there is no clear winner when it comes to tool execution time and memory consumption [80].

Any protocol to be validated using AVISPA is specified using the High-Level Protocol Specification Language (HLPSL). An HLPSL specification consists of a description of the *principals* (i.e., R, I, C, S and the user in our case), security *goals* of the protocol, and the details of the *session(s)* to be analyzed. AVISPA integrates four

Table 4.1: *Summary of AVISPA analysis*

Phase	AVISPA goal*	Coverage
I. Reader-card auth.	Secrecy of K'_{RC} $C \rightarrow R \mid N_C$ $R \rightarrow C \mid N_R$ $S \rightarrow C \mid N_S$	SR1, SR7
II. User auth.	Secrecy of PIN $C \rightarrow U \mid PIN$	SR1, SR7
III. Session-key est.	Secrecy of K'_{RI} $S \rightarrow C \mid N_S$ $I \rightarrow R \mid N_I$ $R \rightarrow I \mid N_R$ $I \rightarrow S \mid N_I$	SR1, SR6, SR7
IV. Main Phase	Secrecy of CMD, ANS $S \rightarrow C \mid N_S$	SR1, SR3

* $A \rightarrow B \mid N$: A authenticates B based on value N
Secrecy of N: Confidentiality of value N is ensured

back-end engines that provide different types of automatic analysis of an HLPSL specification [170]. The tool helps in detecting vulnerabilities against Man-in-the-middle and replay attacks. It also detects whether the HLPSL specification is executable, i.e., all the specified protocol states are traversable. Using AVISPA, we can also optimize our protocols by removing certain parameters from the messages in order to reduce communication overhead and analyze if this results in a new vulnerability.

The analysis of IMDfence using AVISPA is summarized in Table 4.1. The handshake-specific protocol requirements (SR1, SR3, SR6 and SR7) are satisfied by specifying the appropriate goals. In phase III, S extracts user privileges from $Cert_C$ after successful authentication of C, based on N_S in m_{SC_2}. I then verifies S based on N_I to complete the chain from the card to the implant in order to ensure access control. In order to check non-repudiation using the tool, the server verifies that the retrieved sig from the IMD originated from C during the session corresponding to N_S.

4.4.1.2 Reader-specific attacks

When considering all possible attack scenarios, we define the following reader types:

1. *Valid R (R_{valid}):* This is a legitimate device, which is *not* reported as stolen.
2. *Stolen R (R_{stolen}):* A legitimate device which is *reported* as stolen.
3. *Hacked R (R_{hacked}):* A stolen reader which is also modified by A in order to e.g., replace the signature or CMD.

Table 4.2: *Enumeration of attack scenarios Sn in terms of user-reader combinations*

	Reader			
	Valid	Stolen	Hacked	Forged
Trusted, honest user	S1	S2	S3	S3
Trusted, malicious user	S1	S2	S4	S5
Attacker	S6	S2	S7	S7

4. *Forged R (R_{forged}):* A custom-built or software-defined radio used by A in order to communicate with an implant. This reader does not have any pre-shared keys with S.

The following scenarios are possible in terms of user-reader combinations (which are also summarized in Table 4.2):

S1 – Any user & R_{valid}: This is the most common scenario, which must be handled by IMDfence. A cannot insert a false signature remotely (in order to frame someone) since the connection between R and C is protected by MAC-based integrity checks. Moreover, an insider attack (from a legitimate, malicious user) should be detected by the non-repudiation check. However, after sending a malicious command, such a user can attempt multiple harmless write commands in order to eventually overwrite the signature corresponding to the malicious command. We term this as the *signature-overwrite attack*. For each command, 72 bytes of flash space is required to store the signature and the associated session parameters. As an example, if a 32-kB flash memory is allocated for signature storage, 456 attempts will be required to successfully overwrite the targeted signature, which is highly impractical. Even if the user manages to achieve this, the signature record will still point to an abnormally high number of write commands corresponding to a single session, which will raise suspicions.

S2 – Any user or attacker & R_{stolen}: No individual will be able to use R_{stolen} because of the checks involved in the reader-card-authentication phase.

S3 – Trusted, honest user & R_{hacked}/R_{forged}: In order to frame someone, A has to force the legitimate user to use a hacked reader, which replaces the command with an incorrect one. As a guideline, R must be issued from a trusted repository, which rules out the use of R_{hacked} and R_{forged} for trusted users.

S4 – Trusted, malicious user & R_{hacked}: Legitimate malicious users can cover their tracks by using a hacked reader that can replace the signature corresponding to a malicious command, which is to be stored in the IMD, with the one corresponding to a safe command. Such an attack is quite costly to execute and is time-critical since it will involve colluding with someone who has advanced engineering skills while requiring that R_{hacked} is not reported as stolen. Since, the user

is considered trusted by the patient and can thus be in close proximity, he/she has far easier and inexpensive means to harm the patient without getting caught.

S5 – Trusted, malicious user & R_{forged}: Such a user cannot send commands using a forged reader in an online case since R_{forged} does not share a key with S. In the offline case, however, such a user can use a forged reader that is able to create a bogus sig and hence does not require any involvement of C. Moreover, he/she can use the OOB-pairing interface because of being considered as trusted by the patient. Similar to S4, such a scenario also requires hiring an advanced attacker to develop such a reader, and based on the touch-to-access assumption, the user has significantly easier methods to harm the patient.

S6 – Attacker & R_{valid}: For online access, the security protocol will break if A gets hold of a valid reader, card and its associated PIN, accesses the IMD from within the hospital and during the user's working hours, *and* C is not reported as stolen. It is recommended that the user protects her card and PIN, or immediately reports it in case it is lost. Moreover, as a guideline, the user should never lend or sell R to a third party. The protocol will also break if A gets hold of an *OOB-paired* reader and a card with valid respective *tokens*, *and* knows the PIN. We assume that the paramedic resets the pairing after treatment. Overall, A cannot effectively launch the above attacks since the likelihood of all the dependencies being true is extremely low.

S7 – Attacker & R_{hacked}/R_{forged}: For online access, A will not be able to use R_{hacked} because of the reasons mentioned in S6 above. Similarly, A cannot use R_{forged} since it does not have a shared key with S. Moreover, for an offline scenario, getting hold of these readers will not help an attacker A since the session key (K'_{RI}) comes from I in the OOB pairing process. Hence, to gain advantage using these readers, A would still need to get close to I (touch-to-access).

4.4.1.3 Smart-card-specific attacks

Since IMDfence employs smart cards, it is important to ensure that it is safe from the weaknesses [176, 177] present in another widely used smart-card system: EMV (Europay, Mastercard, and Visa). These vulnerabilities exist due to the availability of less secure options for backward compatibility and due to a problematic threat model, in which the reader (i.e., the POS terminal) is assumed to be uncorrupted.

One major issue is that most of the important data is exchanged in plain-text (e.g., account data, amount etc.) since the terminal and the card do not share a symmetric key. Moreover, in the offline use of the cards that do not support public-key cryptography, the PIN is also sent as plain-text. An attacker can modify the unencrypted initialization messages to force the terminal to use this mode [177]. The PIN can be recorded using e.g., a hacked terminal that has additional probes to read data from the smart card interface. In case of an offline-encrypted PIN, the terminal can be hacked to record the keystrokes. Using the account data and PIN, the

attacker can create a magnetic-strip card for use in a country that does not support chip-based smart cards [3].

Another issue is that the terminal cannot use MAC to authenticate messages from the card since they do not share a symmetric key. Cards following the Combined Data Authentication (CDA) scheme from EMV address this by employing signatures. However, in the schemes prior to CDA, the terminal is unable to verify the authenticity of all the card messages either due to unavailability of signatures (in the case of Static Data Authentication, SDA) or the signature-less transaction messages (in the case of Dynamic Data Authentication, DDA). As a result, an SDA card can be cloned for use in offline transactions [177], and a stolen DDA card can be employed in a *two-card attack*, in which the attacker uses his/her own card for PIN verification and uses the stolen card in the transaction phase [6]. Moreover, the card response at the end of PIN verification is unauthenticated. As a result, this response can be modified to deceive the terminal into assuming that the entered PIN is correct.

All these attacks exist because in EMV some of the critical data is left unencrypted or not signed. In contrast, in both the online and offline modes of IMDfence, all data between R and C is encrypted and is authenticated using MACs. Additionally, our recommendation to avoid magnetic-strip-based cards rules out cloning. Similarly, avoiding contactless cards removes an additional attack vector.

Another far more advanced type of attack is the *relay attack* [3, 176], which exploits the fact that the card users cannot know for sure if the display of the terminal is showing correct information. It is a time-critical attack where two transactions are simultaneously taking place. The victim inserts his/her card in a counterfeit terminal (e.g., at a restaurant), which is connected to a fake card of the attacker that is inserted in a valid terminal (e.g., at a jewelry store). The details of the fraudulent transaction are forwarded to the victim's terminal. Her screen shows the correct information, but in effect she pays the amount for the other party.

We observe that the relay attack is far less likely in the case of IMDfence since it requires a legitimate user operating a forged reader. This corresponds to scenario S3 discussed in Section 4.4.1.2.

4.4.1.4 Selection of T

The touch-to-access principle guarantees that an unreasonably high T (reader-card-authentication lifetime) value does not cause a security vulnerability in IMDfence, as evident from Section 4.4.1.2. However, the careful reader may have noticed that a prolonged offline operation enabled by such a large value may result in R's and/or IMD's firmwares becoming outdated. On the other hand, a very small value hinders legitimate access, i.e., availability. Therefore, the hospital server should ensure that T is assigned an appropriate value (within maximum and minimum limits) based on the patient's location and the reader-IMD usage patterns.

Regarding the patient's locality, the probability of having stable Internet connectivity is higher when the patient is based in an urban area compared to a rural setting. Moreover, it stands to reason that the chances of attacker presence ought to be higher in an urban environment. Hence, it makes sense to assign a lower T value for urban areas compared to rural environments. When assigning the T value, reader-IMD usage patterns should also be taken into consideration, which depend on the patient condition and IMD type, ranging from critical implants, such as cardiac defibrillators, to less critical ones, such as neurostimulators. The IMDs requiring frequent reader access should be granted a larger T value. Further investigation on this topic is interesting but is considered outside the scope of this work.

It should be noted that the (re)setting of T can be performed throughout the operational lifetime of the IMD. The physician is required to manually modify this parameter (in S) based on the above guidelines, which then ultimately take effect in the reader-card authentication phase (see Figure 4.3).

4.4.2 Availability – DoS protection

As highlighted in Section 4.1, one of the system requirements is to ensure that the IMD is always *available* for treatment, i.e., it should be protected against DoS attacks. As discussed in Chapter 3, *function DoS* is tackled by employing a dual-CPU paradigm, whereas *battery DoS* is tackled by employing an energy-harvesting-based *zero-power defense* (ZPD) scheme.

In order to assess the viability of IMDfence under energy-harvesting conditions (be it in single- or dual-CPU configuration), we construct the following experimental setup:

(I) Computational costs: Similarly to Chapter 3 (Section 3.5.2), we employ an ARM Cortex-M0+ based 32-bit MCU [153]. Due to its ultra-low-power capabilities, and the on-board hardware-accelerated, security building blocks (i.e., encryption, MAC, hash function, random-number generator etc.), this MCU is becoming increasingly employed in IoT and WBAN settings [24], and hence, is a plausible choice for this evaluation. The security-related computations, i.e., authenticated encryption (AES-128), cipher-based MAC and random-number generation were performed using the MCU's dedicated peripherals ("CRYPTO" and "TRNG"); thus, in our energy measurements, hardware-accelerated primitives are considered. However, as a reference, we also include a software-only MCU implementation of IMDfence.

(II) Wireless-communication costs: Commercial transceiver ZL70103 specifically designed for IMDs has been used [109]. To get reasonable energy costs for (encrypted) data transmission, we chose packet-size lengths similar to the ones used in low-cost RFID tags, due to their similarities with IMDs in terms of computational, memory and energy constraints [164]. Hence N, ID, CMD and ANS were set to 32, 96, 32 and 64 bits, respectively. The sig size was set at 384 bits, which corresponds to

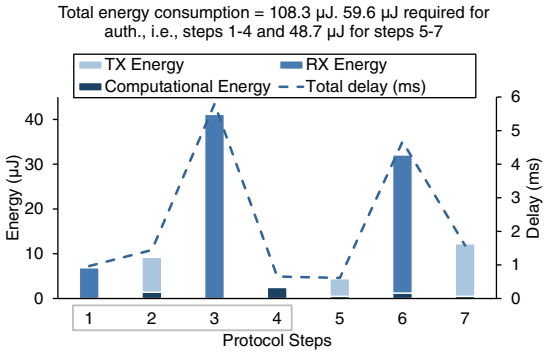


Figure 4.10: *IMD energy consumption and performance per IMDfence-protocol step while using hardware-accelerated security primitives*

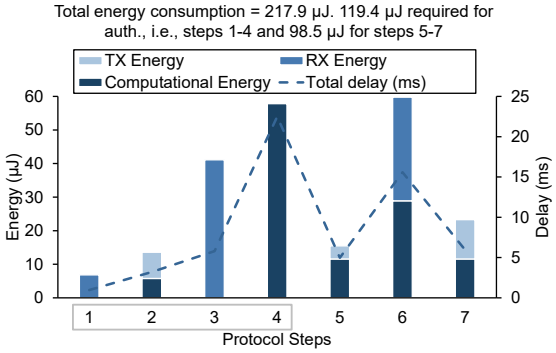


Figure 4.11: *IMD energy consumption and performance per IMDfence-protocol step when implementing the security primitives in software*

an ECDSA (Elliptic-Curve Digital-Signature Algorithm) signature with a 96-bit security level.

The protocol sequence executed by the IMD is shown as numbered steps in Figures 4.5 and 4.6. In the case of hardware-accelerated primitives, the energy consumption for these steps is shown in Figure 4.10. The supply voltage, MCU clock frequency and the TRX data rate are the same as described in Chapter 3 (see Table 3.3). We observe that the energy required for authentication (E_{auth}), i.e., for steps 1 to 4 in Figure 4.5, is only 59.6 μJ. In the case of software implementation, however, E_{auth} is only 119.4 μJ, as shown in Figure 4.11. For such a low harvested-energy requirement (E_{auth}), it has been demonstrated before in Chapter 3 (Section 3.5.3) that real-time performance is possible in the IMD, with or without hardware acceleration. Total IMD energy consumption per type of activity is also shown Figure 4.12.

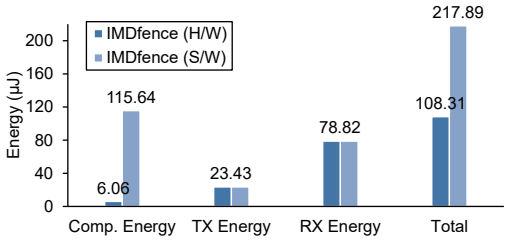


Figure 4.12: *IMD energy consumption per IMDfence-protocol activity*

4.4.3 IMD lifetime

In the previous section, we discussed the feasibility of IMDfence under energy-harvesting conditions to defend against battery-DoS attacks. In this section, we wish to assess the total energy costs that the IMDfence protocol incurs over the whole lifetime of a modern IMD. To do so, we need to consider realistic usage patterns of actual devices, drawn from medical practice. There are two prominent IMD classes: neurostimulators and cardiac implants. Neurostimulators typically consume more power than cardiac devices [106] and, therefore, often come with rechargeable batteries, which would pose no challenge for IMDfence. Cardiac implants, on the other hand, are not rechargeable due to their critical nature (as discussed in Chapter 3), and represent more *pessimistic* devices to assess IMDfence against. Thus, for our evaluation here, we consider a communication session between a pacemaker and a commercial bedside reader (Merlin@home^{TM}) [159].

We consider different data volumes being transferred between the reader and IMD, ranging from a daily three-minute[4] communication session to a three-minute weekly session. Since this reader is intended for monitoring the IMD status, it is assumed that most of the communicated data is transferred from the implant to the reader (e.g., in the form of data logs). Hence, the size of ANS is increased from 64 bits (for a basic session) to roughly 3 MB in order to form a three-minute session. However, for worst-case analysis, the transceiver is considered to be enabled throughout this session and we do not assume the use of energy harvesting for ZPD. Moreover, without loss of generality and in order to more accurately (and pessimistically) quantify the cost of adding IMDfence to an existing system, we consider a dual-CPU IMD, as discussed in the previous section. In this configuration, the security CPU is assumed to execute the complete IMDfence protocol, while the assumed medical-CPU duty cycle and pacemaker stimulation energy are the same as stated in Table 3.3.

With the above consideration, the impact of IMDfence on IMD-battery lifetime can be visualized using Figure 4.13 for different implantable-grade battery sizes [40].

[4]This corresponds to an encrypted session. An equivalent unencrypted session will take roughly half the time due to less data transferred.

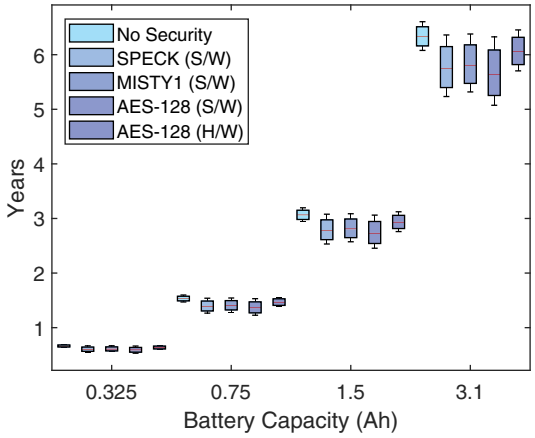


Figure 4.13: *IMD-battery lifetime with respect to cryptographic primitive used. Box-plot variation is due to different data-transfer volumes*

The variability in each data point captures the different volumes of data transfer between the reader and IMD.

Since the majority of the cryptographic operations in the protocol (authenticated encryption and MAC) are based on symmetric block ciphers, as shown in Figures 4.5 and 4.6, it is very interesting to investigate the impact of different cipher versions and/or implementations thereof on IMD lifetime, e.g., a pacemaker. More box plots have, thus, been added to Figure 4.13, where we readily notice that the hardware implementation of AES-128 significantly outperforms the software AES-128 implementation, plus other *lightweight* software ciphers such as SPECK and MISTY1. It is also interesting to observe that the energy impact of the hardware AES-128-based protocol is not significant when comparing with an unsecured communication.

4.4.4 IMD performance

To study the impact of IMDfence on performance during normal operation, we will only analyze the bottleneck of the reader-IMD system in this regard, i.e., the IMD itself. This is because modern readers, such as tablets [162], have far superior computational resources (and battery autonomy) than implants. As far as the smart card is concerned, the amount of computations performed by it is approximately the same as that in commercial uses (e.g., EMV), which we know to exhibit adequate performance.

As far as the IMD is concerned, the performance figure of merit that is crucial to capture here is the delay that IMDfence incurs to the system, both for security computations and data transmission over the air. For unsecured data transfer, the wireless transceiver incurs a delay of 2.2 ms. As shown in Figure 4.10, for (hardware-accelerated) secure data transfer the time delay incurred by each (numbered) pro-

Table 4.3: *Summary of costs for running the IMDfence protocol on an IMD*

	Energy		Delay	Program-memory footprint**
	1 basic session (μJ)	1 daily IMD cycle* (J)	(ms)	(kB)
Without security	16.61	16.60	2.17	16.50
IMDfence (H/W)	108.31	17.69	15.73	24.72
IMDfence (S/W)	217.89	19.89	58.99	26.98

* Which includes a daily three-minute comm. session (see Section 4.4.2)

** This includes the comm. data handling, security processing and MCU peripheral support library for GPIO and USART, which are needed to communicate with the transceiver.

tocol step is no higher than 6 ms, for a total protocol delay of 15.7 ms. Therefore, for the time scales involved in biological processes, we can safely assume that the IMDfence delay overhead is negligible.

4.4.5 Summary of introduced overheads

Table 4.3 summarizes the impact of IMDfence on an IMD in terms of energy, performance and program-memory footprint. For the hardware implementation of IMDfence, it can be observed that, although the energy requirements increase by more than 6 times for a basic session, the *total* daily IMD consumption (that includes a three-minute communication session and electrical-stimulation costs) increases from 16.60 J to just 17.69 J, which amounts to a mere 6.57% increase, as previously shown in Figure 4.13. The reason for this small increase is that the basic medical functionality, e.g., the continuous electrical stimulation of a pacemaker, dominates the security provisions since the reader accesses are far less frequent. In the case of software (AES-128) implementation of IMDfence, the total daily IMD consumption increases by 19.82% (as shown in Figure 4.13). Moreover, there is a minimal increase in the computational delay and required program-memory size. In the context of current MCU technology, 8.22–10.48 kB of additional memory size is negligible. Hence, we conclude that there is no noticeable change in the IMD costs when IMDfence is employed.

4.5 Related work

From the perspective of the research community, we observe a steep rise in the number of works proposed over the last few years [132]. For data confidentiality, integrity and message authentication, the use of lightweight primitives has been proposed. Early works focused on basic security protocols based on symmetric ciphers, which rely on a common pre-shared key between the reader and the IMD [164].

However, such approaches are not scalable in terms of adding new readers that can access the implant. They also do not allow paramedic access during emergencies. Therefore, most of the existing works deal with emergency access, in addition to entity authentication and key exchange. For entity authentication, these works rely on the touch-to-access policy. We now present a brief overview of the latest works from literature that were specifically tailored for IMDs.

Bu et al. propose a low-energy IMD-security scheme called Bulwark [23], which, in addition to satisfying SR1, also allows IMD access in emergencies (SR4). This emergency access scheme is based on Shamir's *secret sharing*, which relies on the users (including the paramedics) to register with the manufacturer of the specific IMD in advance in order to retrieve the access key in case of an emergency. As evident, such a requirement inhibits IMD access in case the patient is out of town (SR8).

Chi et al. [29] propose a protocol that relies on the patient's smartphone for the reader access. However, requiring the patient to be in possession of this additional device (i.e., the smartphone) all the time, including during emergencies, puts a significant burden on the patient.

Belkhouja et al. [15] propose a symmetric crypto system in which they use a *Chaotic key generator* that is employed by both the reader and IMD to generate the symmetric key. However, in order for this key generator to work, both entities are required to have similar pre-installed initial conditions/values. Hence, this scheme cannot function in an emergency scenario, or when the patient is traveling, since the IMD and the reader will not be sharing the same initial conditions.

Wazid et al. [181] and Mao et al. [90] propose three-factor protocols, which rely on passwords, smart cards, and biometrics. Their protocols rely on a reader-registration phase before the IMD deployment in the field. This inhibits SR4 and SR8 since it is unlikely for the paramedic/doctor to possess a pre-registered reader during an emergency or when the patient is visiting abroad. Rathore et al. [126] propose a scheme in which the identifiers of each user (including the patient) are derived from their cardiac signals and are stored in the implant. Hence, it requires a user-registration phase similar to the above protocols. However, their scheme allows emergency access since the paramedic can measure patient's cardiac signal, which is compared by the IMD against the stored identifier in order to grant access. The three-factor protocol from Fu et al. [52] also provides emergency access. However, the patient is required to always be in possession of a personal smart card so that the paramedic is able to use it during an emergency.

As discussed in Chapter 3 (Section 3.4), a few works [27, 43, 62, 87, 164, 186] have also focused on the IMD availability (SR2). In these works, RF energy harvesting is employed to protect the IMD against battery-depletion attacks. In addition, quite a few authentication and emergency-access schemes have been proposed recently that rely on static biometrics (such as fingerprints) [188], dynamic biometrics (such

Table 4.4: *Overview of related works*

	[23]	[29]	[15]	[181]	[90]	[126]	[52]	[43]	[25]	[116]
Confidentiality & Integrity (SR1)	●	●	●	●	●	●	●	●	●	●
Availability (SR2)	○	○	○	○	○	○	○	●	○	○
Non-repudiation (SR3)	○	○	○	○	○	○	○	○	○	●
Emergency Access (SR4)	●	●	○	○	○	●	●	●	●	●
Multi-manufacturer support (SR5)	○	○	○	○	○	○	○	○	○	○
Access Control (SR6)	●	●	○	●	●	●	●	○	○	●
Authentication (SR7)	●	●	●	●	●	●	●	●	●	●
Flexibility & Scalability (SR8)	○	●	○	○	○	○	○	●	●	○
Beside-reader operation (SR9)	○	○	○	○	○	○	○	○	○	○

●: Satisfies requirement, ○: Does not satisfy requirement

as cardiac signals) [25, 43] and combination of both [16]. The interested reader can refer to [4, 26, 132, 183, 187] to get an overview of prior works in this area.

Overall, the above works address only parts of the IMD security requirements (SR1, SR2, SR4, SR6, SR7 and SR8), which is also summarized in Table 4.4. For instance, non-repudiation is not considered and the emergency-access schemes do not take into account the (current) multi-manufacturer environment. To the best of our knowledge, there is no protocol that provides all the services highlighted in Section 4.1.

The work from literature that came closest to fulfilling the above requirements was proposed by Park [116]. It establishes a session key between the IMD and a *personalized* reader based on shared secrets between these entities and a trusted third party (hospital server). The use of public-key crypto in the personalized reader and the server facilitates non-repudiation. However, the work lacks a few additional pieces in order to properly close the non-repudiation gap. The protocol addresses access control by first allowing only read access to the implant via the server. Based on the result of the read-out data, the server provides write keys to the reader-IMD pair which allows the user to change IMD settings. The personalization process involves the physician inserting a personal smart card into the reader. However, since it resembles a single-factor authentication for the user (i.e., through the use of a smart card without PIN), any person in possession of a valid (stolen) card can access the implant by getting hold of a reader. The server maintains a list of primary-care physicians authorized to access each registered implant. If the physician is a member of this list, then a read-key is granted to the physician. We believe that maintaining such a user list is not scalable, it inhibits flexibility, and hence, should not be employed. As an example, such a scheme will not work in case the patient requires some treatment at a hospital abroad. Besides, the proposed emergency-access scheme uses a bracelet that has a secret key. However, such token-based security schemes are single points of failure (e.g., in case the token is stolen or the contents are disclosed). Also, it requires the patient to wear the bracelet at all times, which is inconvenient. Moreover, in the emergency scenario, the scheme drops access control and non-repudiation. Lastly, this work excludes battery DoS from its adversarial model, and it does not consider bedside-reader operation.

4.6 Summary

In this chapter, we have proposed a novel security protocol for IMD ecosystems, IMDfence. We have demonstrated that our approach offers a meticulous coverage of security requirements that are critical to these systems. This becomes possible through the use of a personal smart card and a trusted third party, which helps in facilitating access control, non-repudiation, user authentication, bedside-reader operation and system scalability. We have also shown that IMDfence does not introduce any noticeable overheads in the implant, and it has the ability to support

zero-power defense against battery-DoS attacks. It is observed that our proposed protocol increases the total IMD energy consumption by just 6.57%, which is minimal in the context of the IMD lifespan. We have also proposed an OOB-channel-based version of IMDfence, which enables offline or emergency access.

CHAPTER 5

"There are only two types of companies: those that have been hacked, and those that will be."

FBI Director Robert Mueller, 2012

Secure device pairing and zero-power defense using ultrasound waves

M. A. Siddiqi, R. H. S. H. Beurskens, P. Kruizinga, C. I. De Zeeuw, and C. Strydis, "Securing Implantable Medical Devices Using Ultrasound Waves," *IEEE Access*, vol. 9, pp. 80 170–80 182, 2021.

M. A. Siddiqi, C. Strydis, and C. I. De Zeeuw, "Implantable Medical Device and Control Device Therefor," Jun. 29 2021, NL Patent App. N2028563 & N2028564.

In this chapter, we will provide further insights into the concept of OOB-based device pairing, which was introduced in the previous chapter. Most of the device-pairing schemes from literature are based on the *touch-to-access* policy (see Section 2.2.5.1). However, these schemes rely on the authenticator, i.e., the IMD, periodically polling for the requester over the untrusted wireless channel to kick-start the pairing process *before* proximity is established. This makes the IMD susceptible to battery-DoS attacks (see Chapter 3). As a result, existing device-pairing schemes still require the use of energy-harvesting[1]-based ZPD (EH-ZPD) to protect against battery DoS. However, energy harvesting requires additional components next to the transceiver, such as a harvesting circuit, power management and an energy reservoir (see Figure 3.2), which increase design complexity. It also has to satisfy additional frequency-band and medical-safety constraints in order to be used in an IMD.

In this chapter, we propose SecureEcho, an ultrasound-based device-pairing scheme that protects against battery DoS without actually implementing energy harvesting, which reduces the associated design complexity. SecureEcho achieves secure pairing by using ultrasound as a body-coupled-communication (BCC) channel for sharing a cryptographic key. The completely passive nature of the proposed circuit allows the IMD communication interface to remain asleep before any access is made via the BCC channel, which enables ZPD. To the best of our knowledge, ultrasound has never been used for key transport in *plaintext* before. This is because of the absence of an in-depth security evaluation of this channel, as inferred from the various works in literature [99,123,124]. Therefore, in this chapter, we also provide a comprehensive security evaluation of this channel in order to prove its robustness against eavesdropping and message-insertion attacks.

This chapter, thus, makes the following novel contributions:

- A lightweight device-pairing security protocol that utilizes ultrasound in order to protect against battery-depletion attacks.
- A comprehensive security evaluation of ultrasound as an *inherently secure* BCC channel.
- A proof-of-concept implementation and validation of the SecureEcho approach.
- A detailed comparison of SecureEcho and the traditional energy-harvesting-based ZPD method.

The rest of the chapter is organized as follows. Background on the BCC concept is provided in Section 5.1. We explain our proposed reader-IMD device-pairing

[1]We will use the term *energy harvesting* for both RF- and inductive-coupling-based harvesting techniques.

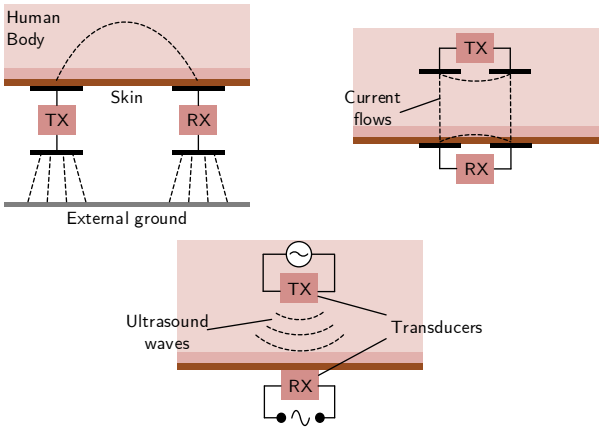


Figure 5.1: *General types of BCC: Capacitive coupling (top left), Galvanic coupling (top right) and Ultrasound communication (bottom)*

scheme, SecureEcho, in Section 5.2. In Section 5.3, we mount a comprehensive security evaluation of the ultrasound BCC channel and, in Section 5.4, we provide the proof-of-concept implementation of our approach. A detailed comparison of SecureEcho and the EH-ZPD approach is provided in Section 5.5. Section 5.6 reviews the related work. We draw overall conclusions in Section 5.7.

5.1 Background

Over the past decade, numerous touch-to-access schemes have been proposed to securely pair the reader and the IMD. These schemes can be categorized as: Biometric-based, Proxy-based, Token-based and Distance-based schemes (see Section 2.2.5.1). The SecureEcho scheme described in this chapter falls under the last category. In such schemes [77, 124], weak or OOB signals are employed to exchange secrets or keys, or determine the distance between the devices in order to be sure of proximity.

Please note that in this chapter, we will use the term *direct (or plaintext) key transport* when a symmetric key is sent in plaintext from one entity to another over an OOB channel. We will use the term *key agreement* when both entities exchange public-key material over the OOB channel, which is then used to compute the symmetric key.

5.1.1 Body-coupled communication

There is an emerging trend of using the human body as an OOB channel not only for reader-IMD pairing, but also for pairing devices within a wireless body area network (WBAN) [173]. Three general, body-coupled communication (BCC) techniques are capacitive coupling, galvanic coupling and ultrasound communication, respectively, as shown in Figure 5.1. In capacitive coupling, the signal propagates

through the body from a transmitter electrode to the receiver in the form of electromagnetic waves while the return path between the two nodes is formed by electrostatic coupling between their second electrodes and an external ground [129]. In the case of galvanic coupling, the transmitter sends the signal through the body by inducing alternating current into the tissue, which is received by the two receiver electrodes [129, 182]. In ultrasound, a piezoelectric or a capacitive transducer at the transmitter side converts an electrical signal into acoustic waves (at frequencies $>$ 20 kHz), which are detected by a similar transducer at the receiver and converted back into the original electrical signal [135].

The external return path of capacitive coupling results in electromagnetic leakage, which can be sniffed by an attacker [129,173]. As a result, it can only be used for *key agreement*, i.e., exchanging the *public* keys, and not for key *transport* in plaintext. Galvanic coupling, is more localized and has been used for direct key transport in [95]. A preliminary security evaluation of this channel in [84] indicates that it is secure against attacks from distances $>$ 0.5 m. However, its authors still recommend a comprehensive analysis that also takes into account different transmit powers and antenna gains of the attacker device.

Ultrasound can also potentially be used for direct key transport. However, to the best of our knowledge, such a work does not exist in literature. This will be discussed in detail in Section 5.1.2.

It should be noted that the use of BCC for the whole reader-IMD communication session, instead of just the key establishment, is *impractical* due to its very nature. For example, it is not possible to have regular communication with a bedside reader that is a few feet away from the patient. Hence, switching to an RF transceiver is necessary in order to support long-range telemetry.

5.1.2 Ultrasound communication

Ultrasound has been proposed as a BCC channel for data transfer in quite a few recent works, such as [72, 76, 134, 135]. It has also been proposed as a wireless-power-transfer (WPT) channel for recharging IMDs [14, 165]. Furthermore, it is being touted as an in-body communication and WPT channel for next-generation, mm-sized neural implants for both the Central (CNS) and Peripheral Nervous Systems (PNS) [140, 185]. This is because the size of ultrasound transceivers can be several orders smaller than their electromagnetic (EM) counterparts, which is ideal for scaling-down of IMDs. Moreover, the power attenuation of ultrasound waves in soft tissue is significantly smaller than that of EM waves, leading to deeper tissue penetration and relaxed medical-safety constraints [14, 185]. However, to the best of our knowledge, its applicability in secure data transfer (e.g., direct key transport) has not been pursued. This is mainly due to the lack of evaluating the security of this channel.

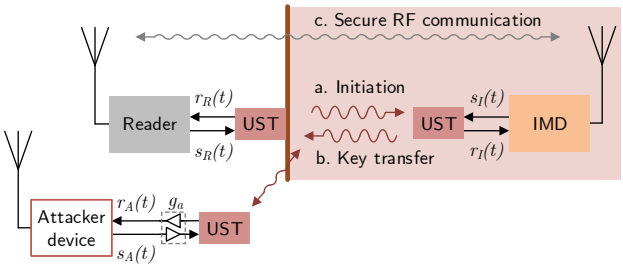


Figure 5.2: *BCC-based reader-IMD pairing. UST: Ultrasound Transducer*

5.2 SecureEcho device pairing

In this section, we present our device-pairing scheme, SecureEcho, which is tailored to protect IMDs against battery-depletion attacks in addition to establishing trust with an external reader.

SecureEcho employs ultrasound as a BCC channel. Although this scheme can work with *either* ultrasound or galvanic coupling, we prefer the former. This is because the ultrasound transducers offer highly directional and very-short-range communication depending on the frequency of operation and transducer width, which is ideal for secure key transport. The security evaluation of this channel will be discussed in detail in Section 5.3.

5.2.1 System and attacker model

We build on the system and attacker models described in Chapter 2 (Sections 2.2.1 and 2.2.2). In addition, we assume that the ultrasound-receiver circuit of the IMD is purely passive in nature, i.e., it does not consume any additional energy. This will be important for the discussion pertaining to message-insertion attacks.

5.2.2 Security protocol

The idea behind our scheme is briefly summarized in Figure 5.2. To pair a reader with an IMD, the ultrasound probe of the reader is first placed on the patient skin surface at a point closest to the implant. This is because the ultrasound propagation range is very short for MHz-range transducers and the acoustic absorption in air is very high. Since only a trusted person is able to come this close to the patient, which involves touching the skin for a prolonged period of time, this type of access can be considered strongly in line with the touch-to-access principle. The IMD can, thus, now safely assume that the message received from the ultrasound channel is from a trusted entity. Assuming that this channel is secure from eavesdropping, which we will discuss in detail in Section 5.3, the IMD can securely transport a symmetric key, which can be used to secure the subsequent RF communication.

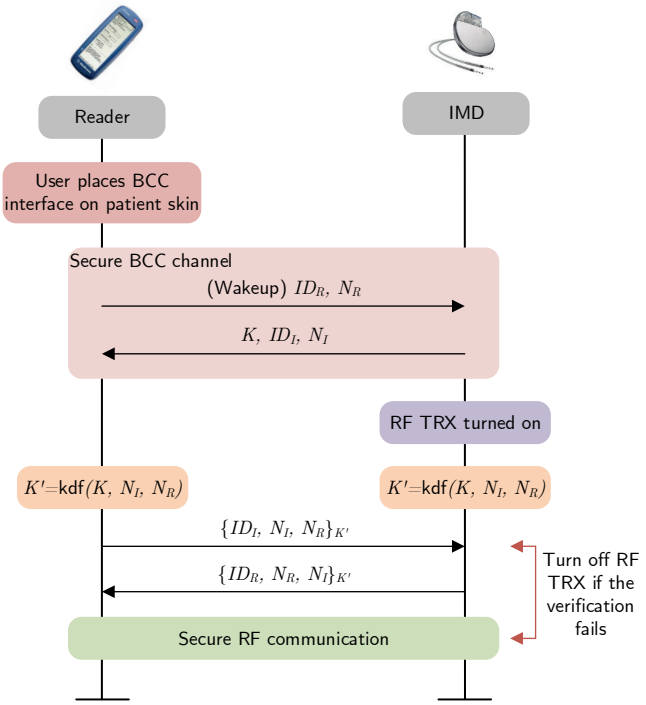


Figure 5.3: *Reader-IMD protocol for initial pairing*

The above secure device pairing can be achieved by following the protocol in Figure 5.3 (refer to Table 2.1 for the employed notations). The reader sends an initiation message via the ultrasound channel in order to wake up the implant and start a communication session. This message contains a randomly-generated nonce (N_R) and the reader identifier (ID_R). The IMD responds with its own identifier (ID_I), nonce (N_I) and most importantly, a fresh and random long-term key (K). The IMD then turns on the RF transceiver for data communication. Both entities calculate a short-term session key $K' = \mathsf{kdf}(K, N_I, N_R)$ to be used for encrypting subsequent messages, where $\mathsf{kdf}()$ can be any secure key-derivation function. The reader then sends the nonces and ID_I as an encrypted message over the RF channel. The IMD decrypts and verifies the received message to be certain that the other entity is authentic and is in possession of K'. If the verification fails, the IMD turns off the RF transceiver and aborts the protocol. Otherwise, it sends the nonces and ID_R as an encrypted message to the reader.

The reader decrypts the message received from the IMD and verifies its contents. At this point, both entities have mutually authenticated each other. A secure communication channel between the two entities has been established, and hence, they can now proceed with encrypting the subsequent messages using K'.

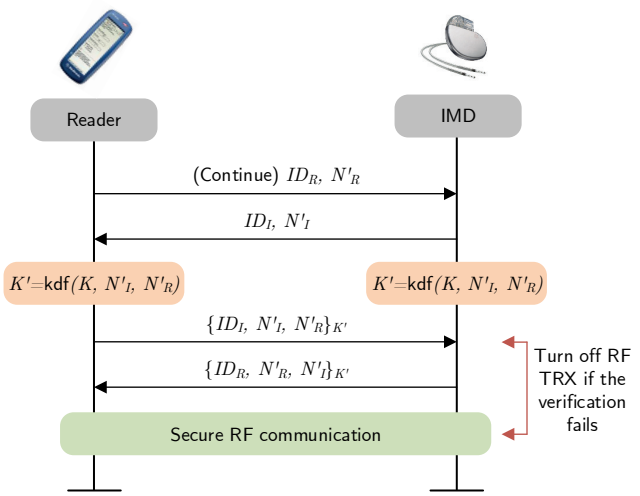


Figure 5.4: *Secure communication protocol over the RF channel based on a pre-shared long-term key K*

The key point during this pairing process is that the RF transceiver can *only* be woken up by the IMD MCU/processor. Since the attacker is unable to use the ultrasound BCC channel without the patient noticing, RF communication can never happen, and hence, battery DoS cannot be launched.

For the subsequent sessions, i.e., when both the devices already share a long-term key, the initial pairing is *not* required. In this case, the protocol from Figure 5.4 is executed over the RF channel using fresh nonces (N'_I and N'_R), which is based on the three-pass mutual authentication protocol specified in ISO/IEC 9798-2. In case a MAC check fails at the IMD side or when the received nonces and identifier do not match, e.g., in the case of a battery-DoS attack using bogus messages, the IMD turns off its RF transceiver and exits the protocol. For the next legitimate access, the devices would then again be required to undergo the pairing of Figure 5.3.

5.2.3 System architecture

Figure 5.5 shows the overview of the proposed system architecture. There is a separate MCU/processor for executing the medical application (*medical MCU*), and for handling communication packets and running the security protocol (*security MCU*). As discussed in Chapter 3, this dual-processor architecture, which is based on [164], protects against function DoS: If an attacker sends continuous packets to prevent the IMD from running its main application, only the security MCU will be kept busy entertaining those messages, whereas the medical MCU will remain unaffected. However, in order to protect against battery DoS specifically, additional measures are required, as explained below.

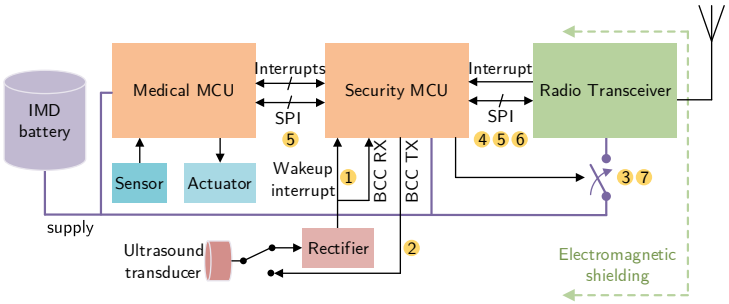


Figure 5.5: *SecureEcho system schematic along with the numbered steps of the security-MCU finite state machine*

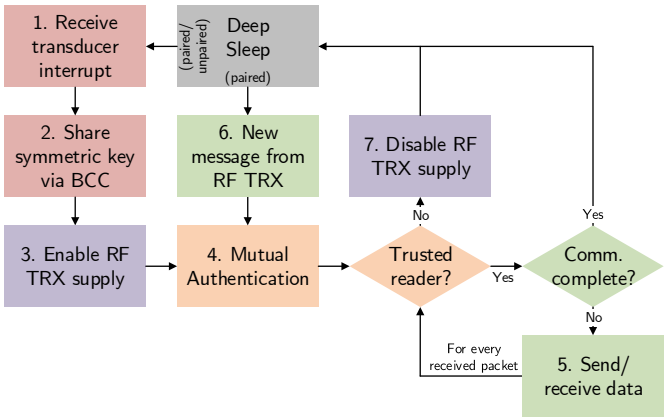


Figure 5.6: *State machine of the secondary, security MCU*

In the default (unpaired) state, the RF transceiver is *powered off* and the security MCU is in its lowest power or *deep-sleep* state. The finite state machine (FSM) of this MCU is shown in Figure 5.6. It is important to note that, during this unpaired state, the RF transceiver does not wake up periodically to check the presence of an incoming RF signal. The security MCU is first woken up from its deep sleep via the ultrasound interface. In order to achieve *true* ZPD, this interface is required to operate passively, i.e., without consuming any additional energy. Fortunately, an ultrasound transducer can do just that; it can passively convert incident waves into an electrical signal so that it can be used to wake up the security MCU. This will also be demonstrated in Section 5.4.

The IMD will use this interface to transport the long-term key K, as previously shown in Figure 5.3. The security MCU will then signal to power up the RF transceiver. The IMD is now ready to receive encrypted RF packets. When the communication session is over, the RF transceiver will go to sleep (instead of get-

ting powered off) and, similarly to commercial IMDs, it will periodically wake up to check (or *sniff*) for an external entity trying to communicate with the IMD. In the above or any of the future sessions, if a packet authentication fails, the security MCU will reset the pairing by turning off the RF transceiver. Hence, in order to start the communication with the IMD again, the reader would be required to repeat the ultrasound BCC pairing.

The system architecture also includes an electromagnetic-shielding cage, which protects against side-channel attacks (to be discussed in Section 5.3.3). The RF antenna lies outside this cage so that (secure) RF communication is not affected.

5.3 Security analysis of ultrasound communication

To perform a comprehensive security evaluation of the ultrasound communication channel, which we employ in SecureEcho, two ways exist: (1) Physical-setup based and (2) simulation based. Regarding the first approach, it would be too cumbersome and impractical to perform the security analysis on an actual setup while taking into account the different variables, such as the transducer frequency, attacker distance, directivity etc. As a result, we follow the second approach instead, in which we employ acoustic simulations using the open-source *k-Wave toolbox* [174] built in MATLAB. k-Wave is an increasingly popular and well-studied simulation tool for modeling acoustic wave-field propagation in heterogeneous media. k-Wave efficiently solves a system of first-order, coupled equations that accounts for phenomena such as acoustic absorption and complex tissue-wave interactions that play a part when waves are transmitted through the skin and other layers. Moreover, k-Wave has been validated experimentally and it has become one of the standards for accurate and fast ultrasound simulations [60, 97].

The acoustic properties of the media encountered in an IMD setting and employed in k-Wave simulations are taken from [63, 180] and are summarized in Table 5.1. The acoustic impedance ($Z = \rho c$) and absorption coefficient (α) values significantly contribute to the attenuation of the ultrasound signal. When the signal travels from one medium (medium 1) to the next (medium 2), then the *transmission coefficient* (i.e., the ratio of the transmitted-signal amplitude and the incident-signal amplitude) is $2Z_2/(Z_1 + Z_2)$ [113]. For $Z_1 >> Z_2$, the signal will experience a very-high attenuation. In addition, these waves suffer absorption at a rate of α dB/m, which increases with frequency.

The transducer efficiency for the simulations is set to 3.8 kPa/V (1 kPa = 1000 Pascals [N/m^2]) in order to match the one used in our proof-of-concept design (see Section 5.4). The resulting acoustic intensities in W/m^2 (based on the employed signal voltages in our study) are well within the FDA safety limits for ultrasound operation [48]. For the digital data transfer over the ultrasound channel, ASK modulation (on-off keying) with non-return-to-zero (NRZ) data encoding is employed. These schemes are used to simplify the analysis without loss of generality. We ran

Table 5.1: *Acoustic properties for different media encountered in an IMD scenario*

Medium	Speed of sound, c (m/s)	Density, ρ (kg/m^3)	Acoustic impedance, Z (kg/m^{2}s $\times 10^6$)	Absorption coefficient, α (dB/m)[§]
Air	346	1.2	0.0004	161
Gel[*]	1480	1000	1.48	0.16
Skin	1624	1109	1.801	129.95
Fat[**]	1477	911	1.345	42.99

[§] At 1 MHz
[*] It acts as a coupling medium between the skin and the external probe.
[**] Subcutaneous Adipose Tissue (SAT)

the simulations using three different transmit frequencies, 0.5, 1 and 2 MHz, which are used in WPT schemes and ultrasonography, to find a secure range of operation.

5.3.1 Passive (eavesdropping) attack

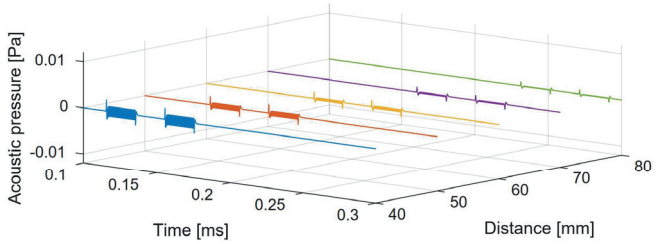
We first investigate whether an attacker can successfully eavesdrop on the key K, which is transported via the ultrasound channel. For this test, we assume that the IMD applies a 3.3 V (amplitude) signal to its transducer. This voltage level is consistent with the batteries used in such devices.

Figure 5.7 shows the acoustic attenuation of ASK-modulated bits (1, 0, 1, 0) with respect to transducers of different resonant frequencies, at a bit rate of 50 kbps. We notice a significant attenuation with the increase in the transducer resonant frequency. This is mainly because the acoustic absorption increases with frequency. This already gives us an indication of the improbability of retrieving the signal correctly after a few centimeters at frequencies $\geq$ 2 MHz.

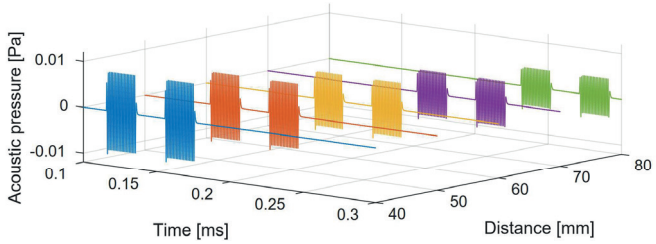
To analyze this concretely, we perform a bit-error ratio (BER) analysis of the ASK-demodulated signal with respect to the attacker's distance and the employed transducer (frequency). The received signal $r_A(t)$ at the input of the attacker's demodulator is given in (5.1), where $s_I(t)$ is the source (modulated) waveform that drives the transducer at the IMD side (see Figure 5.2). $h(t)$ is the overall impulse response of the acoustic medium and g_a is the voltage gain of the attacker's receiving amplifier. $n_t(t)$ is the thermal noise due to the transducer and $n_a(t)$ is the noise introduced by the receiving amplifier.

$$r_A(t) = g_a \cdot \{h(t) * s_I(t) + n_t(t)\} + n_a(t) \tag{5.1}$$

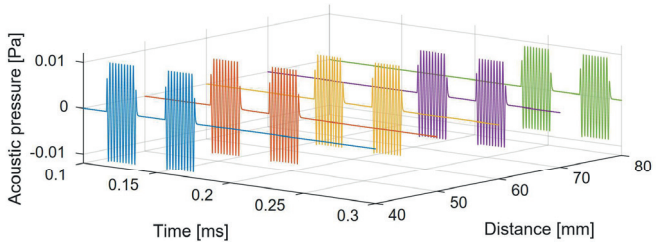
The RMS value ($\bar{n}_t$) of $n_t(t)$ is shown in (5.2), where k_B is the Boltzmann constant, R is the transducer resistance tuned to the amplifier's input resistance, T is the temperature, and Δf is the transducer bandwidth.



(a) *2 MHz*



(b) *1 MHz*



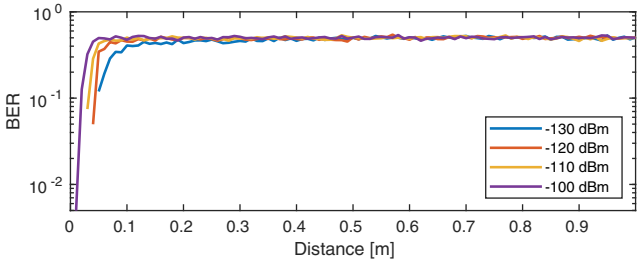
(c) *500 kHz*

Figure 5.7: *Acoustic-signal attenuation over distance for different transducer resonant frequencies*

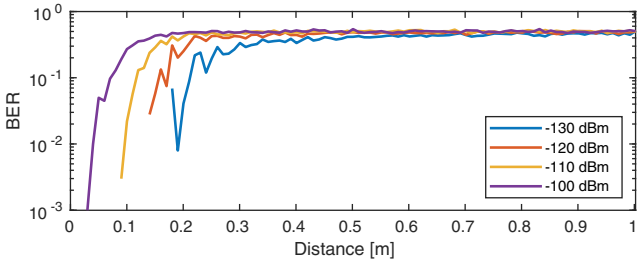
$$\bar{n}_t = \sqrt{4k_B T R \Delta f} \tag{5.2}$$

Then, to see the effects of both the noise components ($n_t(t)$ and $n_a(t)$) on the demodulated signal, we use the overall noise floor N_{dBm}, which is calculated using (5.3).

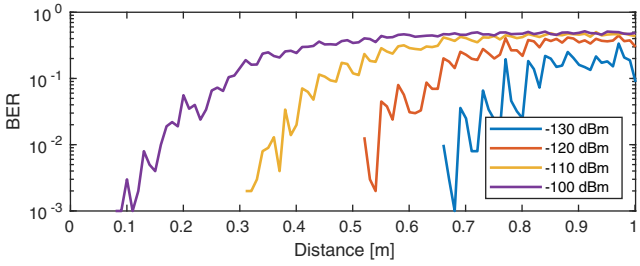
$$N_{dBm} = \underbrace{10 \cdot \log_{10}\left(\frac{\bar{n}_t^2}{4R}\right)}_{\text{due to the transducer}} + \underbrace{10 \cdot \log_{10}\left(1 + \left(\frac{\bar{n}_a}{\bar{n}_t g_a}\right)^2\right)}_{\text{amplifier noise figure}} + \underbrace{10 \cdot \log_{10}(1000)}_{\text{dB to dBm conversion}} \tag{5.3}$$



(a) *2 MHz*



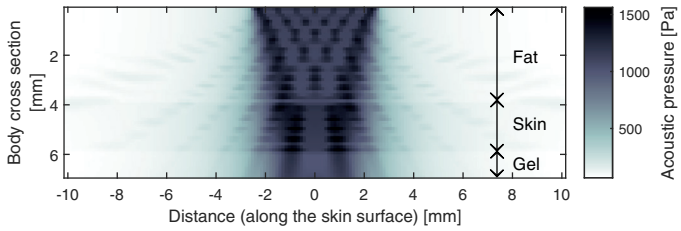
(b) *1 MHz*



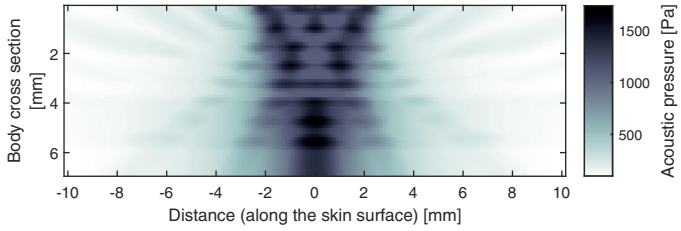
(c) *500 kHz*

Figure 5.8: *Bit-error ratio BER over distance with respect to different noise-floor levels*

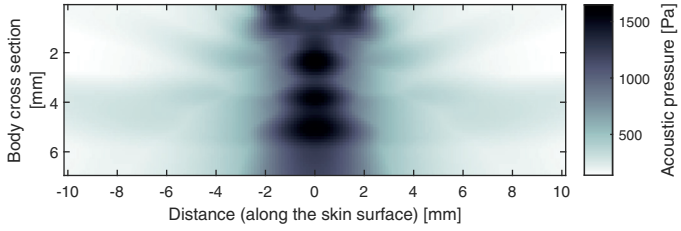
We assume that the attacker is using an advanced, very-high-gain and very-low-noise receiver. Since the N_{dBm} of the receive chain depends on the exact implementation, we provide the BER plots with respect to a range of noise floors (see Figure 5.8). As a reference, for a 2-MHz ultrasound transducer with a 1-MHz bandwidth and its resistance tuned to 50 Ω, and an example advanced amplifier [51] having a 50-Ω input resistance, an input noise of 2.3 nV/$\sqrt{\text{Hz}}$ and a 60 dB gain, the overall noise floor ≈ -114 dBm at 20 °C. From Figure 5.8, it can be observed that for a digital acoustic signal originating from the IMD, successfully demodulating it over the air medium for a 2-MHz transducer is not possible beyond 5 cm. For a 500-kHz transducer, the eavesdropping range increases to around 60 cm for an extremely-low -130 dBm noise floor. This analysis indicates that the eavesdropping attack is



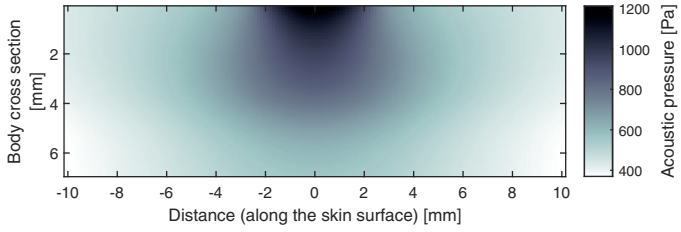
(a) *2 MHz*



(b) *1 MHz*



(c) *500 kHz*



(d) *100 kHz*

Figure 5.9: *Directivity tests for 5-mm width transducers.*

physically unrealistic to launch when using a transducer with a resonant frequency in the MHz range.

We now perform the eavesdropping analysis from a different perspective, i.e., by considering the impact of the ultrasound-wave directivity, which primarily depends on the transducer frequency and width. This analysis allows us to confirm that even if the attacker is very close by, the directivity needs to be maintained in order to successfully eavesdrop.

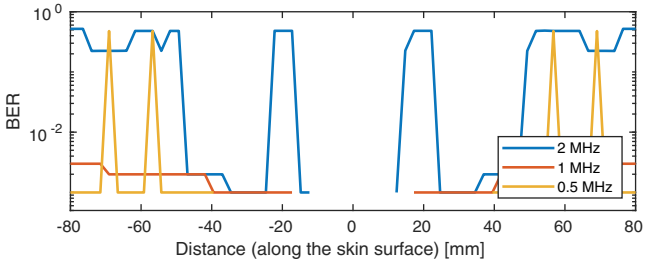


Figure 5.10: *BER of the received acoustic signal along the skin with the assumed noise floor of -130 dBm.*

For a transducer 5 mm wide, which is among the typical sizes used in WPT and BCC [14], the directivity plots are shown in Figure 5.9. After emanating from the transducer, the signal first traverses through layers of fat (4 mm), skin (2 mm) and ultrasound gel (1 mm) before entering the air medium. Although such layers are simulated, k-Wave is used by professionals in the ultrasound field to accurately assess material attenuation due to its advanced numerical model. We can observe that the transducers of MHz-range frequencies are highly directional. This is also supported by the BER plots with respect to the distance along the skin, i.e., along the direction parallel to the face of the IMD transducer, as shown in Figure 5.10. The BER worsens if the alignment is disturbed by even a couple of centimeters. These tests show that, in addition to being *very* close, the attacker also has to maintain a strict line-of-sight alignment with the IMD transducer. Even a subtle movement of the patient, e.g., when they are breathing, will cause disruption in the eavesdropping.

5.3.2 Battery-DoS and active attacks

Since the IMD employs a passive ultrasound receiver with no amplification, the onus is on the attacker to pre-amplify (with gain g_a) the input signal, $s_A(t)$, of their transducer (see Figure 5.2) so that the DC level of the wakeup/received signal ($r_I(t)$) at the IMD side is greater than the logic level '1' threshold (V_{thr}) of the IMD-MCU's GPIO pin. For a worst-case approximation (best case for the attacker), we only include the effects of acoustic *attenuation* and do not consider acoustic-signal *distortion* since it is irrelevant when the aim of the attacker is to overcome V_{thr} at the IMD. As a result, $h(t) \approx g_{ch} \cdot \delta(t - \tau)$, where g_{ch} is the overall transmission coefficient of the heterogeneous acoustic medium, defined in (5.4), and τ is the introduced delay.

$$g_{ch} = 2^{n-1} \cdot \prod_{i=1}^{n-1} \frac{Z_{i+1}}{(Z_i + Z_{i+1})}, \ \forall\, n \in \mathbb{Z}^+ \mid n > 1 \tag{5.4}$$

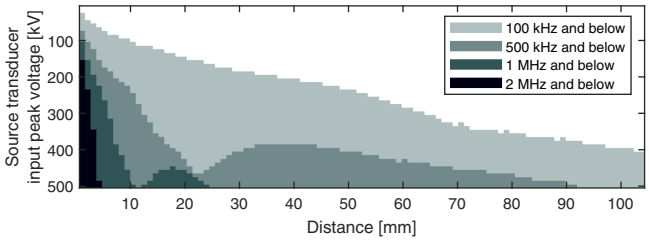


Figure 5.11: *Supply voltages required by the attacker in order to successfully launch a battery-DoS attack over air with respect to different resonant frequencies and distances. A non-white grid point represents a successful attack.*

Here, n is the number of medium changes the acoustic signal undergoes during transit. Based on the above approximation and (5.1), the attacker then has to satisfy (5.5) in order to successfully launch an active (message-insertion) attack.

$$|g_a \cdot g_{ch} \cdot s_A(t-\tau) + n(t)|_{max} > V_{thr} \tag{5.5}$$

However, as shown by the acoustic simulations in Figure 5.11, it would require an unrealistically high signal amplitude to launch a successful attack. For even a slight air gap, the attacker would need to apply a few hundreds of kilovolts at the source transducer, which is not practical at all. The reason for this is that g_{ch} from the (attacker) transducer to air is $\sim 2.6 \times 10^{-5}$ for a Lead-Zirconate-Titanate (PZT) transducer, which is insurmountable in the absence of any amplification at the IMD side. To make matters worse for the attacker, the directivity discussion from Section 5.3.1 applies here as well.

5.3.3 Side-channel attacks

It has been shown [57, 124] that it is possible for the acoustic circuit to get a signal from the RF receiver chain due to interference, effectively resulting in the reception of an unwanted acoustic signal. This phenomenon can lead to active signal-injection attacks from the adversary. However, this can easily be prevented by adding electromagnetic shielding over the ultrasound circuitry [56, 124], which is addressed in the system architecture (see Section 5.2.3). This shielding also prevents the electromagnetic signals (corresponding to the signals driving the IMD transducer) to leak out of the IMD, which protects against the potential eavesdropping.

5.3.4 Summary

In this section, we demonstrated through realistic simulations that ultrasound BCC is sufficiently secure when using a transducer that is sensitive to frequencies ≥ 1 MHz. Based on our analysis, it can be concluded with certainty that the attacker would not be able to successfully launch eavesdropping, message-insertion and

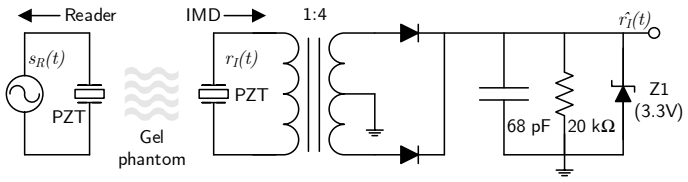


Figure 5.12: *Rectification circuit for the proof-of-concept implementation*

battery-DoS attacks: They would need to get really close (within a few millimeters), maintain directivity, and in the case of message-insertion and battery-DoS attacks, would need to bring impractically-large-sized equipment on site. The analysis is also valid for even more conservative threat models [94] that assume the attacker being able to get close to the patient in a *crowded* place.

5.4 Proof-of-concept implementation

In Section 5.3, we concluded that the ultrasound transducers of MHz frequencies are sufficiently secure. In this section, we will practically demonstrate that such transducers can be actually used for secure device pairing and ZPD.

For our proof-of-concept design, a 2.25 MHz ultrasound PZT transducer from Panametrics (model: V306) [113] with a transmit efficiency of 3.8 kPa/V is employed. The same 32-bit ultra-low-power MCU that was used in Chapter 3, is employed as the IMD security MCU.

The BCC receive path consists of a PZT and a rectification circuit (see Figure 5.5), which is also used for generating the wakeup signal: The high-frequency sinusoid at the output of the PZT is rectified into a digital (demodulated) signal, which is connected to the MCU BCC-RX and wakeup-interrupt pins. The transmit path, on the other hand, is much simpler: the MCU BCC-TX pin is directly connected to the PZT (similarly to [135]). In this case, the MCU performs the ASK modulation by generating a 2.25 MHz signal using its internal high-frequency-RC oscillator for a bit-period duration to represent a '1'. The absence of this signal represents a '0'.

As discussed in Section 5.2.3, the rectification circuit has to be passive in order to achieve *true* ZPD. As a result, the amplification of both the transmit and receive signals has to be done at the reader side. However, this is not problematic since the power constraints at the reader are sufficiently relaxed compared to the IMD.

The rectifier schematic is shown in Figure 5.12, which is designed so that the reader can communicate and wake up the implant when the ultrasound probe is placed on the body at the point closest to the IMD. We used four different mediums/phantoms between the source and receiver PZTs (see Figure 5.13 and Table 5.2). The best-case medium (in terms of maximum acoustic-energy transfer) was the

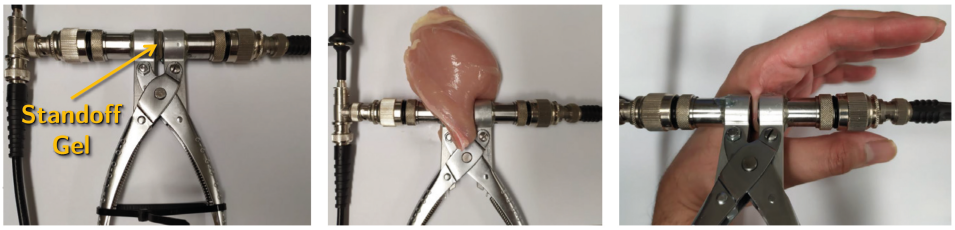


Figure 5.13: *Mediums/Phantoms employed: Standoff Gel (left), chicken breast (center) and human hand (right)*

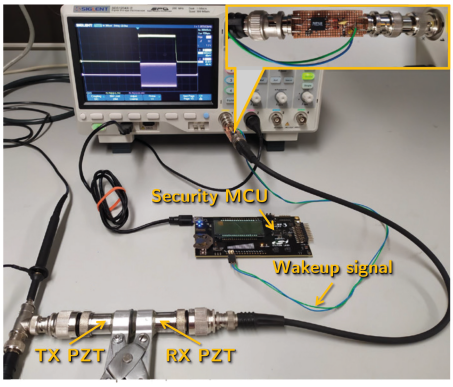


Figure 5.14: *Experimental setup*

Table 5.2: *Measurements from the implementation setup*

Medium/Phantom	Thickness (mm)	Min. required peak $s_R(t)$ (V)	Dissipated[§] TX power (W)	Dissipated TX Energy* (mJ)
Standoff Gel	6	32.5	0.77	7.88
Chicken breast	5.5	34.5	0.87	8.91
Hand** (subject 1)	4	44	1.41	14.44
Hand (subject 2)	5	39	1.11	11.37

§ Acoustic power transferred through the medium from the reader
* At a data rate of 50 kbps for a packet size of 512 bits
** Adductor-pollicis-muscle region (between the index finger and the thumb)

homogeneous standoff gel, whereas the worst-case[2] mediums were the adductor-pollicis-muscle regions (between the index finger and the thumb) of two subjects. Figure 5.14 shows the proof-of-concept implementation setup. Figure 5.15 shows the oscilloscope snapshot of the input to the reader PZT, $s_R(t)$, and the resulting

[2]This medium has two skin layers which results in more acoustic losses compared to an actual case, such as an implanted pacemaker, in which there is one skin layer (in addition to fat) between the reader and the IMD.

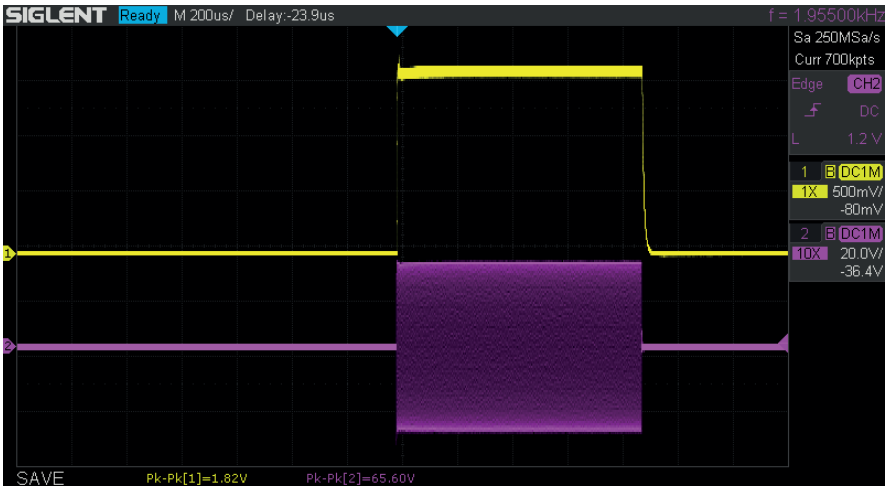


Figure 5.15: *Oscilloscope snapshot of the $s_R(t)$ (magenta) and $\hat{r}_I(t)$ (yellow) signals.*

IMD rectifier output, $\hat{r}_I(t)$. The small number of components in the rectifier allows it to easily fit in any IMD class, along with the above-mentioned PZT. Table 5.2 lists the minimum required peak voltages of $s_R(t)$, which result in the $\hat{r}_I(t)$ having a DC level $\approx$ 1.8 V, which can successfully wake up the MCU and also represent a logic level '1' when receiving data. It can be seen that the TX energy transferred through the medium is small enough (i.e., less than 15 mJ) to fall within the budget of a battery-powered portable reader. It should be noted that the calculated acoustic power transferred through the medium and that too for a small duration of time, i.e., for sending ID_R and N_R (see Figure 5.3), is comfortably within the FDA safety limits [48].

5.5 Comparison with EH-ZPD

We now compare SecureEcho with the traditional energy-harvesting-based ZPD (EH-ZPD) approach. An overview of this comparison is provided in Table 5.3. Next, we will go over the comparison points, which are derived from the ZPD design considerations in Chapter 3 (Section 3.3), one by one.

5.5.1 Frequency-band and safety constraints

As discussed in Chapter 3 (Section 3.3.3), for an EH-ZPD design, separate bands should be used for reader-IMD communication and energy harvesting in order to satisfy the FCC constraints while also supporting reasonable data rates. Since SecureEcho does not require energy harvesting, it does not need a separate band for WPT, which eases up frequency-band constraints. Moreover, the medical-safety constraints imposed by the FDA are relatively easier to meet in the case of ultrasound

Table 5.3: *Comparison of SecureEcho with EH-ZPD*

Design consideration	SecureEcho	EH-ZPD
Frequency-band constraints	+	−
Medical-safety constraints	+	−
Operating range (bedside-base-station operation)	+	−
Emergency access	+	−
Design suitability	+	−
Dependability	+	−
Secure device pairing	+	−
Device usability	−	+
Energy overheads	+	+

+/−: relatively good/poor performance

compared to EH-ZPD since the limit for ultrasound power transmission into the tissue is higher compared to that of electromagnetic power transfer [14].

5.5.2 Operating range

SecureEcho allows the use of a bedside base-station reader after its initial BCC pairing with the IMD. On the other hand, in the case of EH-ZPD, harvesting RF energy over the bedside range (a few feet) requires larger antennas/coils, and longer delays due to the charging of the energy reservoir, which complicates the IMD design (see Chapter 3).

5.5.3 Emergency access

In the case of a paramedic access to the IMD in an emergency scenario, one main requirement is to provide trust establishment between the reader-IMD pair without any pre-shared secret between the two entities. This is reasonable to assume because in emergencies, the paramedic reader and the patient IMD are likely unknown to each other. SecureEcho inherently provides this feature since the secret (symmetric key) can be transported securely using the ultrasound channel. Moreover, since this transfer requires a physical contact, it satisfies the touch-to-access assumption. On the other hand, an IMD with EH-ZPD cannot establish trust on its own, and therefore, would still require a pairing mechanism.

5.5.4 Design suitability

The EH-ZPD architecture has many moving parts in addition to the transceiver, such as a harvesting circuit, power management and an energy reservoir (see Figure 3.2). On the other hand, an ultrasound-coupling-based BCC transceiver is much simpler (as demonstrated in Section 5.4). This gives it an advantage in terms of *design suit-*

ability, i.e., the tedious approval cycle of such a ZPD module is likely to be much shorter than a harvesting-based design.

5.5.5 Dependability

5.5.5.1 Reliability

Related to the discussion in Section 5.5.4, since SecureEcho has a lower number of electronic components, it aids in *dependability* since each such component has an associated failure rate. This is important to consider for safety-critical systems, such as IMDs.

5.5.5.2 Maintainability

In the case of EH-ZPD, since the authentication is executed using free energy, the harvesting circuit and the energy reservoir (such as a supercapacitor) have to be designed according to the required authentication energy. It is possible that, in the future, the employed cryptographic primitives may require replacing (via over-the-air firmware updates) due to newly found vulnerabilities. However, this may require the replacement of the harvesting circuitry as well, which is not possible for an already implanted device. This is not a problem for SecureEcho since the BCC circuit is agnostic to the employed cryptographic primitives.

5.5.6 Secure device pairing

In general, in the absence of a trusted-third party, for any two devices requiring key-exchange (for supporting confidentiality, integrity and authentication), they need to perform asymmetric (or public-key) cryptography. Public-key cryptography is also required if the devices need to support non-repudiation (see Section 4.1.2). To protect against man-in-the-middle (MITM) attack, which is a common attack against public-key cryptography, the devices require the use of certificates and a public-key infrastructure (PKI). However, when it comes to IMDs, they only have a limited on-board memory, which is problematic for storing necessary certificates, and they lack an Internet connection, which is required to track the validity of all possible reader certificates [96]. One way of getting around the need for certificates is for the IMD to verify that the reader is in close proximity [131, 148], or in other words, enforce the touch-to-access principle. Similarly to what was discussed regarding emergency access above, SecureEcho inherently ensures proximity between the reader and the implant, which is not the case for EH-ZPD, as it would still require a touch-to-access scheme.

Related to above, SecureEcho can act as a robust pairing method (or in other words, *association model*) for existing communication standards like Bluetooth LE, which is increasingly being employed in modern reader-IMD systems. Bluetooth LE offers four association models: *Just Works*, *Passkey*, *Numeric comparison* and *OOB*

(out-of-band) pairing [22]. Just Works does not offer MITM protection, whereas the passkey and numeric comparison require a user interface on the device (e.g., a touch screen), which is not possible for an implant. OOB pairing is an ideal association model for Bluetooth-LE-enabled IMDs, and SecureEcho can slot in as an OOB channel with minimal modifications.

5.5.7 Device usability

In terms of device usability, the main difference between SecureEcho and EH-ZPD is that the former requires a water-based ultrasound gel to be applied on the skin before the initial pairing. However, this is *not* required for subsequent accesses between the already paired devices. Moreover, the initial gel application can be considered as acceptable given that such a practice is already prevalent in ultrasonography.

5.5.8 Energy overheads

The SecureEcho pairing is only employed infrequently, since the devices that are already paired do not need to repeat it. As a result, the additional energy overhead introduced by SecureEcho has a negligible impact on the IMD lifetime (see Sections 5.5.8.1 and 5.5.8.2 for details). Also, given that EH-ZPD would still require a touch-to-access scheme (as discussed in Section 5.5.3), the overall solution will exhibit similar or higher energy consumption than SecureEcho.

5.5.8.1 Determining energy overheads

The total energy consumption for an IMD that provides basic security (without ZPD) is stated in (5.6). E_{sec} includes the energy consumed by the security computations, data handling and the RF transceiver. E_{med} includes the energy consumed by the medical application, the sensing of physiological signals, and the electrical stimulation applied on the human tissue.

$$E_{total} = E_{med} + E_{sec} \tag{5.6}$$

In the case of SecureEcho, E_{sec} is shown in (5.7). Here, P_{MCU} is the average active-mode power consumption of the security MCU. P_{RF} is the average active-mode power consumption of the RF transceiver. t_{auth} and t_{main} are the durations of the authentication and main (data-transfer) phases, respectively. t_{BCC} is the time taken by the BCC key-exchange. t_{total} is the duration over which the energy is being calculated. Lastly, P_{sleep} is the average sleep-mode consumption of the security MCU and the RF transceiver.

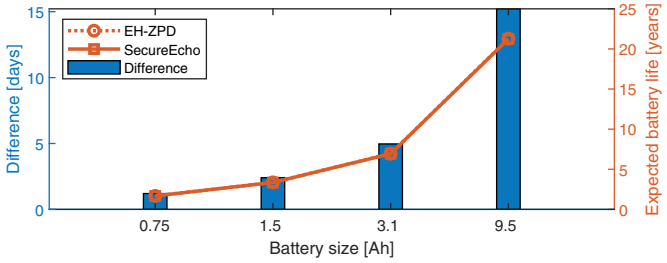


Figure 5.16: *Differences between the expected battery lifetimes when using SecureEcho compared to EH-ZPD*

$$\begin{aligned} E_{sec}^{BCC} &= P_{MCU} \cdot t_{BCC} \\ &+ (P_{MCU} + P_{RF}) \cdot (t_{auth} + t_{main}) \\ &+ P_{sleep} \cdot (t_{total} - t_{BCC} - t_{auth} - t_{main}) \end{aligned} \tag{5.7}$$

For EH-ZPD, in which the authentication phase is executed on free energy, E_{sec} is shown in (5.8).

$$E_{sec}^{EH} = (P_{MCU} + P_{RF}) \cdot t_{main} \tag{5.8}$$

For $t_{total} >> t_{BCC}, t_{auth}, t_{main}$, i.e., over a very long course of time and coupled with the fact that the pairing only has to be done when it is reset (i.e., seldom), and $t_{main} >> t_{auth}$, the overhead introduced by SecureEcho becomes:

$$\Delta E_{sec} = E_{sec}^{BCC} - E_{sec}^{EH} \approx P_{sleep} \cdot t_{total} \tag{5.9}$$

With the lowest-energy-mode currents in modern MCUs getting lower than 100 nA [153], the above overhead has a negligible impact on the IMD lifetime, as discussed next.

5.5.8.2 Impact on battery life

Taking the example of a typical pacemaker, we now calculate the impact of SecureEcho on the IMD battery life compared to using EH-ZPD. We reuse the pacemaker specifications that were earlier summarized in Table 3.3 (Chapter 3). The differences between the expected battery lifetimes, when using SecureEcho compared to EH-ZPD, are shown in Figure 5.16. It is clear that the impact of SecureEcho is hardly noticeable.

5.5.9 Discussion

From the above analysis, it can be concluded that SecureEcho significantly outperforms EH-ZPD, except in the case of device usability because of the minor require-

ment of using a water-based medium or gel *before* the pairing process. However, this is *not* required for subsequent accesses between the already paired devices, i.e., during the normal use of the reader. Moreover, EH-ZPD is dependent on a pre-existing OOB pairing scheme (in the absence of an Internet connection). On the other hand, SecureEcho elegantly provides *both* secure device pairing and ZPD.

5.6 Related work

Mayrhofer et al. [99] performed a high-level threat analysis for ultrasound communication. They assumed that an attacker can eavesdrop on this channel if they are in the same room, and that a line of sight is not required when using this channel. Such an assumption had had to be made since a comprehensive security analysis was not available at the time. Mayrhofer et al. also proposed a method for secretly sending nonces via the ultrasound channel: First, a user ensures that the devices to be paired are aware of the distance between each other. The sender device first sends an RF synchronization message, and then, after a delay, sends an ultrasound pulse. This delay represents the value of the secret (or nonce). The receiving device extracts the message by calculating the delay between the received RF synchronization message and the ultrasound pulse, and subtracting the known distance. In the case of reader-IMD communication, however, the absence of the user interface on the IMD prevents the user from verifying that the two devices have agreed on a correct distance.

Besides, acoustic waves within the *audible* frequency range were employed for direct key transport by Halperin et al. [62]. In this scheme, the IMD sends a random key using this channel and the reader listens to this transmission at a very short range. However, this scheme was soon found to be vulnerable to passive eavesdropping from 5-6 feet away [61].

Rasmussen et al. [124] also proposed an acoustic-channel-based device pairing. However, instead of direct key transport, a distance-bounding scheme was employed. In such a scheme, the IMD calculates the delay between the sent and received transmissions in order to determine the physical distance between the reader-IMD pair. The IMD allows access if the reader is in very close proximity. Its security depends on the fact that an attacker cannot send a message to the acoustic interface faster than the speed of sound in air. One of the main differences of the above solution with SecureEcho is that its acoustic interface is not fully passive, which rules out its use as a ZPD scheme (when not using energy harvesting). Another issue is that this interface employs a band-pass filter, amplifier and a phase-locked loop, which results in a (much) more complex design compared to SecureEcho.

The latest work from Putz et al. [123] proposes an acoustic-channel-based device pairing in which the devices send their *public-key* material via an audio interface. *Integrity codes* are employed to detect whether the keys were modified while in transit. These public keys can then be used to derive a shared symmetric key, e.g., in the

form of a Diffie–Hellman key exchange, in order to secure the RF communication channel. The solution is tailored for pairing devices that already have a built-in audio and user interface, such as smartphones. A user can trigger the start of the pairing process by enabling the acoustic interfaces on both the devices (via the respective applications). However, in the case of reader-IMD systems, the absence of an IMD user interface implies that the pairing startup will require an initial communication between the two devices over an untrusted channel, and for one device to periodically poll for the other. As in the above works, this is done *before* proximity has been established. As a result, the above schemes are susceptible to battery-DoS attacks *if* energy-harvesting-based ZPD is not in place. SecureEcho, on the other hand, provides an elegant solution of inherently providing ZPD without requiring any energy harvesting, as shown in Section 5.2.

5.7 Summary

In this chapter, we have presented SecureEcho, a secure device-pairing scheme for reader-IMD systems that inherently provides protection against battery-depletion attacks. We have shown that the ultrasound channel used in the pairing process is sufficiently secure at MHz-range frequencies. We have also demonstrated a proof-of-concept implementation of the passive circuit that enables the pairing process and ZPD. We conclude that SecureEcho outperforms the traditional EH-ZPD in terms of satisfying frequency-band and medical-safety constraints, operating range, emergency access, design suitability and dependability.

CHAPTER 6

"Left unchecked, technical debt will ensure that the only work that gets done is unplanned work!"

Gene Kim, The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win, 2014

Determining the economic viability of adding security to IMDs

M. A. Siddiqi, A.-A. Tsintzira, G. Digkas, M. Siavvas, and C. Strydis, "Adding Security to Implantable Medical Devices: Can We Afford It?" in *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '21. USA: Junction Publishing, 2021, p. 67–78.

In Chapter 1, we highlighted the *four* reasons behind the IMD industry's slow reflexes in addressing growing cybersecurity concerns. Among them was the prohibitive nature of the expected perpetual costs for IMD-code maintenance due to adding security. This hypothesized reason is the main drive behind this chapter, which looks at IMD security from a fresh angle: *Is it economically sustainable to add security to IMDs?* Convincing the manufacturers (and other stakeholders) that IMD security is economically viable and sustainable, will liberate a lot of security solutions that are considered at the moment "out of scope". Along the lines, it will also allow to invert the psychological bias of denial ("we do not need security") and complexity ("we can only add very trivial security due to costs").

In order to address the IMD-economics question in a tangible way, we adopt in this chapter the systematic concept of *technical debt (TD)* for capturing the software costs of modern IMDs. TD expresses the implied development cost that is incurred due to taking shortcuts during software development in order to reduce the time-to-market of a product. We prefer TD over other methods (e.g., QMOOD [11] or CK [30]) because it covers a large variety of issues, ranging from code-convention violations to architectural problems. On top of that, the monetized nature of TD is proven to be a more helpful way to communicate maintainability benefits to non-software-engineering stakeholders, compared to the traditional software metrics.

Hardware costs, even if feasible to capture via TD, would be mostly irrelevant for IMDs. Due to the IMDs' deeply-embedded nature, hardware changes never occur within a given device's lifetime (since it is implanted) and occur rarely within a given product line. Modern IMDs are, thus, software-driven devices (see Strydis [163, Chapter 2]), meaning that hardware changes incur virtually no TD. Besides, such changes can be captured via their repercussions in the respective software codebase, which changes far more frequently by comparison.

With this chapter, we make the following novel contributions:

- A systematic analysis of security-related software costs in IMDs, based on a synthetic historical record of IMD-codebase changes.
- Predictions of the TD impact of IMD medical and security codebases on future IMD costs.
- Along the lines, a short technical-feasibility study of inserting mainstream security mechanisms in commercial IMDs.

The rest of the chapter is organized as follows. Background on the technical-debt concept is provided in Section 6.1. In Section 6.2, we present our experiment design and, in Section 6.3, we provide the details of the various IMD-software versions developed for this study. TD is calculated based on these versions and evaluated in detail in Section 6.4; future predictions on cost are made. We provide an overview of related works in Section 6.5. We conclude the discussion in Section 6.6.

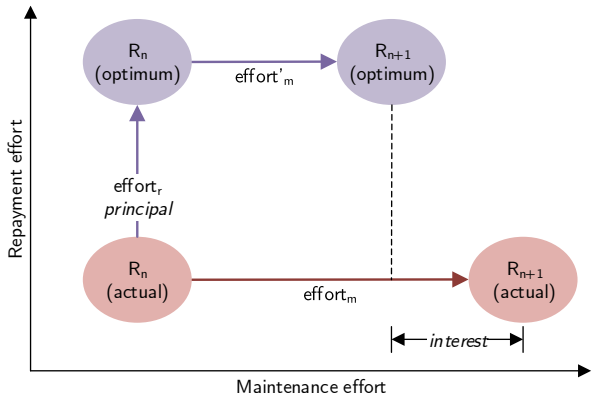


Figure 6.1: *Relationship between TD principal and interest where R_n is the n^{th} design release [28]*

6.1 TD background & employed toolflow

Technical Debt (TD) is composed of two parts: *principal* and *interest*. The principal is the amount of money a company has to pay in order to develop a software system to its optimal quality. If optimal quality has not been reached, this effectively means saving effort, or in other words, money. This amount (i.e., the principal) increases the company's capital, and can be invested in other activities. However, this *internal loaning* comes at a cost: any future *maintenance activity* on the codebase, e.g., for accommodating a new feature, will require increased effort due to the less-than-optimal code quality and maintainability [36]. These additional efforts, are equivalent to paying an interest on a loan. In contrast to the financial interest, which is calculated at regular time intervals based on a given interest rate, TD interest is amassed only when the software artifact is being maintained.

Figure 6.1 illustrates the above concepts. Every design has the potential to reach an *optimal* quality level compared to its *actual* level. In order to reach this level, the development team needs to dedicate some effort, which is equal to the **TD principal**. This activity, which involves code refactorings, is important for the repayment of TD, and hence, it is also called *repayment effort* ($effort_r$). On the other hand, the effort performed in order to add a new feature, enhance functionality or fix bugs is called *maintenance effort* ($effort_m$). In the case of maintaining an optimal design version, adding a feature requires $effort'_m$, whereas in the actual case, the same activity requires $effort_m$, which is always greater. The difference between these two efforts is the **TD interest**. It is important to note that $effort_r$ and $effort_m$ represent only *coding* and *verification* efforts.

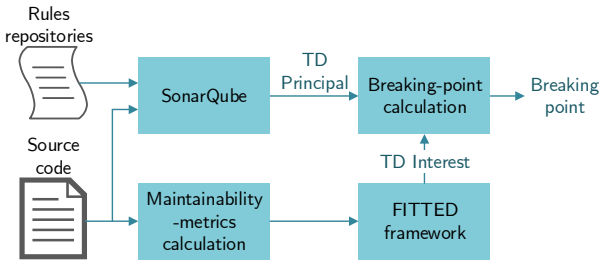


Figure 6.2: *Tool flow employed for TD calculation*

Algorithm 1: TD Principal (P_n) calculation for a design release (codebase) R_n

Inputs : $n, R_n, rules, r_H$
Output: P_n
$N_R \leftarrow size(R_n)$;
$N_{rules} \leftarrow size(rules)$;
$hours \leftarrow 0$;
for $i \leftarrow 1$ **to** N_R **do**
  for $j \leftarrow 1$ **to** N_{rules} **do**
    $V_{i,j} \leftarrow$ total j^{th}-rule violations in the i^{th} file;
    $T_{i,j} \leftarrow$ time required to fix $V_{i,j}$;
  end
  $hours \leftarrow hours + T_{i,j}$;
end
return $P_n \leftarrow hours \times r_H$;

6.1.1 Employed tools

In this work, the different code releases (i.e., intervals) of the IMD application are committed using the Git version-control system. TD principal is quantified via SonarQube, which uses the SQALE method [83] to measure the system TD (see Figure 6.2 and Algorithm 1). This tool first gets its input, i.e., file-level changes between these releases, through JGit, which is a Java library implementing Git. It then (1) reflects the application source code against a set of predefined rules, to identify violating code snippets, and (2) calculates the time required to resolve each violation. The total time required to fix all the violations represents the *TD principal*. This value, which is in *hours*, is converted into *currency* using a standard hourly rate (r_H) of **USD 45.81**, which is in line with the rate of an average developer in the US [175].

The *TD interest* can be calculated in various ways. In this work, it is calculated using FITTED, a framework for managing interest in technical debt [5, 28] (see Figure 6.2), which assesses TD interest by calculating the difference between the efforts

required to *maintain* optimal and non-optimal software artifacts, respectively (see Figure 6.1). The metrics ($\mathbf{m} \in \mathbb{R}^4$) used for quantifying maintainability in order to calculate the above efforts are *coupling*, *cohesion*, *cyclomatic complexity* and *size* (in lines of code). They are calculated using the approach in [8] and are defined in Section 6.1.2. In FITTED, a *fitness* function $f : \mathbb{R}^4 \rightarrow \mathbb{R}$, is employed that takes the above metrics as input and returns the *fitness values* of the actual and optimal software artifacts, which are $f(\mathbf{m}_n)$ and $f(\mathbf{m}'_n)$, respectively, where $\mathbf{m}_n$ and $\mathbf{m}'_n$ are the non-optimal and optimal metrics belonging to the n^{th} code release. $effort_m$ and $effort'_m$ are directly proportional to the respective fitness values. TD interest (I) accumulated between releases $n-1$ and n can then be calculated using (6.1) [28]:

$$\begin{aligned} I_n &= effort_m - effort'_m \\ &= k_n \left(1 - \frac{f(\mathbf{m}'_n)}{f(\mathbf{m}_n)}\right), \ \forall\ n \in \mathbb{Z}^+ \end{aligned} \tag{6.1}$$

Here, k_n represents the lines of code that are added between the current (n) and the previous release ($n-1$).

6.1.2 Metrics definitions

We now briefly describe the metrics referred to in this chapter that are related to both TD principal and interest:

Accumulated TD: It is the sum of TD principal and interest, and thus, represents the total technical debt.

Coupling and Cohesion: Coupling indicates the number of dependencies between the files of a software project. The more the dependencies, the more difficult it becomes to maintain and extend a software system. The maintainability-metrics calculator determines coupling using (6.2), where FO_i (fan out) is the number of files referenced by the i^{th} file, and N_R is the total number of files in a code release.

$$coupling = \frac{1}{N_R} \sum_{i=1}^{N_R} FO_i \tag{6.2}$$

Cohesion represents the degree to which the lines inside a file interact with each other. It is calculated using the *lack of cohesion in methods (LCOM)* metric. If X_i is a set of line pairs in the i^{th} file that do not share any variable, and Y_i is a set of line pairs that have at least one common variable, then $LCOM_i$ can be calculated using (6.3) [30], where the maximum cohesion corresponds to the $LCOM$ value of 0:

$$LCOM_i = \begin{cases} |X_i| - |Y_i|, & |X_i| > |Y_i| \\ 0, & otherwise \end{cases}, \ \forall\ i \in \{1, 2, \cdots, N_R\} \tag{6.3}$$

The overall cohesion of a code release is calculated using (6.4).

$$cohesion = \frac{1}{N_R} \sum_{i=1}^{N_R} LCOM_i \tag{6.4}$$

Low coupling and *high* cohesion are desirable qualities, indicating that the software is easy to understand, maintain and extend.

Cyclomatic complexity (CC): It refers to the number of independent paths throughout the code. Whenever the program control flow branches, e.g., due to an *if* statement, the cyclomatic complexity increases by one. The higher the CC, the harder the software becomes to understand, maintain and test: The larger the number of paths, the more the tests required to achieve a sufficient code test coverage.

Lines of code (LOC): This metric indicates the number of lines of software code that are not part of a comment. LOC can be used to estimate the programmers' productivity, during the development phase, or the software's maintainability during the production phase.

Breaking point: It is the point in the future (in terms of code releases) at which the cumulative TD interest[1] reaches and surpasses the TD-principal amount. At that point, all savings accumulated by not repaying TD will have been exhausted as a result of the additional maintenance effort during the software evolution [28]. The breaking point b_n for the n^{th} code release (R_n) is calculated using (6.5), where P_n is the TD principal at R_n:

$$b_n = \frac{P_n}{\sum_{i=1}^{n} I_i} \tag{6.5}$$

6.2 Experiment design

We now explain our experiment design: In order to perform TD analysis of the IMD application code, we start by constructing a synthetic historical record of IMD design changes, captured as code releases over time, targeting both medical and security aspects of those IMDs. TD is not affected by exact years but, in order to also give readers a precise as possible timeline, this historical record dates back in the past as far as 1997 and extends to speculated future releases until 2028, so as to capture future IMD changes; see Table 6.1. This record will permit us to analyze the impact these software releases have on IMDs in terms of TD amassed.

An *ideal* TD analysis would require all the application-code releases to be coming from the IMD manufacturers. However, there are various obstacles to that approach:

[1] The cumulative TD interest is the sum of the current and all previous TD interests. It should not be confused with the accumulated TD.

Table 6.1: *Overview of the constructed IMD timeline. The year, type of release, design-information sources and hardware modifications (if any) are shown.*

Release	Year	Release type	Description of added design feature	Source(s)	Peripherals added*	
					Medical MCU	Security MCU
1	1997	Medical	Processor-based basic medical functionality	[178]	ADC, Cryotimer**	–
2	1999	Medical	TRX connection for configuration updates	[109, 119]	USART/SPI	–
3	1999	Medical	Read-out of sensor values	[101]	–	–
4	2001	Medical	Battery-level monitoring and read-out	[100]	–	–
5	2002	Medical	Safety modules, e.g., watchdog timer	[168]	Watchdog	–
6	2003	Medical	OTA-firmware-update support	[47, 157]	–	–
7	2008	Medical	Read-out of data logs with time-stamps	[169]	RTC	–
8	2017	Security	Fundamental security services	[103]	–	–
9	2020	Security	DoS-attack protection	[31, 33, 43, 62, 164, 186]	–	2 × USART/SPI
10	2023	Security	Replace SW cipher with a HW implementation	[153]	–	Crypto module
11	2026	Security	New security services	[52, 116, 144]	–	–
12	2027	Medical	Multi-sensory operation	[78]	–	–
13	2028	Security	Secure emergency mode	[43, 52, 132, 144]	–	Cryotimer

'–' : Not applicable or no change compared to previous.
* Core peripherals (e.g., clock- and energy-management units) not included.
** Ultra-low-energy timer of [153].

1. There is no known repository that hosts application code from IMD manufacturers.

2. The sensitive nature of these products, coupled with the traditionally cryptic culture of the IMD industry, has made acquiring code sources directly from the manufacturers virtually impossible.

3. The other potential option is to reverse-engineer explanted IMDs. However – setting aside the ethical, legal and practical hurdles – this method will only give us access to the firmware binaries at best.

The above obstacles necessitate employing a *synthetic* codebase in the sense that the included IMD code has been synthetically created based on publicly available clinician's manuals (from multiple manufacturers), news articles, data sheets, and so on (see source(s) column in Table 6.1). This is a painstaking process and, yet, the only viable means of analyzing the IMD field currently undergoing a critical transition and drawing important conclusions for both the scientific and the industrial communities. Our confidence in the codebase representability is further safe-guarded by (a) employing auxiliary metrics (see Sections 6.4.1 and 6.4.2), and (b) making it publicly available[2] in this work so as to encourage a critical review and improvement by the various IMD stakeholders.

Next, we will go over the IMD classes considered, the selected hardware, and crucial assumptions made in setting up our experiment. A detailed presentation of the IMD code releases will be provided in Section 6.3, which is essential for motivating our results.

6.2.1 IMD applications

Two prominent IMD application classes are considered: neurostimulators and cardiac pacemakers. Cardiac implants hold the largest market share, whereas the neurostimulators are projected to witness the fastest growth. Roughly more than 50% of all IMDs in use belong to these two classes [59]. The hardware and software features included in this work largely capture the characteristics of actual IMDs. These features are inferred from publicly available information, from *multiple* IMD manufacturers, and are a good approximation for answering the research questions raised in this study.

In this work, the general closed-loop structure is kept the same in both the classes. One of the main differences is the sampling frequency (f_s) employed to capture the physiological signal. It has been shown that, for cardiac implants, f_s can be as low as 62.5 Hz [155] whereas, for neurostimulators, an f_s of 100 Hz is sufficient since most brain activity can be found within the 0–50 Hz range [166, 178]. The general structure of the application is based on the lightweight, wavelet-based

[2] `https://gitlab.com/neurocomputing-lab/sims`

filter design presented in [178]. Overall, we have encoded IMD software in C, which is consistent with the state of the art available throughout the assumed time period of study.

Although the doctor's *reader device* or the bedside base-station [159], are crucial components of the broader IMD system as well, in this study we strictly included IMD-application code, for two reasons: (i) IMDs are the bottleneck in terms of resources, e.g., their battery cannot be replaced during the operational lifetime. (ii) Most of the critical attacks, such as battery depletion, have a lasting effect on the IMD operation, and they are not targeted towards the reader.

6.2.2 IMD hardware platform

IMD manufacturers use commercial off-the-shelf (COTS) microcontrollers (MCUs) as their processing and/or controlling cores in modern IMDs [21]. To the best of our knowledge, these manufacturers do not design their own processors. In this work, the IMD-application source codes were tested on an EFM32 Tiny Gecko MCU, which is based on a 32-bit ARM Cortex-M0+ CPU [153] from Silicon Labs. In addition to being ultra-low power, the development kit and the integrated development environment (IDE) of this MCU come with Advanced Energy Monitoring, which enables live and accurate measurement of current draw. MCUs based on Cortex-M have been employed in latest commercial IMDs available, according to the official Bluetooth SIG listing of declared products and qualified designs [21]. Hence, this MCU is a suitable choice for our analysis. Moreover, for the costs associated with the wireless communication, a commercial implantable-grade transceiver, Microsemi ZL70103, has been used [109].

As will be shown in Section 6.4.1, the compiled code fits in the MCUs that were commercially available throughout the assumed time period of study. Moreover, the different application versions conform to the processing capabilities of such MCUs. The starting date for our analysis corresponds to the year when the 16-bit TI MSP430 – known to be used in IMDs – was first released [13, 132]. Although MCUs and microprocessors started appearing in commercial IMDs long before the MSP430 (e.g., RCA 1802 used in [82]), they did not come with C compilers. Since our TD analysis is only possible on applications written in C, MSP430 is an early enough and realistic starting point of our assumed timeline.

The above make it obvious that our hardware setup remains fixed throughout our experiments. This does not pollute our evaluation process since, as discussed in the beginning of this chapter, hardware-caused TD is negligible. Conversely, pinning down the hardware platform used, allows for an even comparison of the different code releases. Finally, it should be noted that the low-level, peripheral-support library provided by the MCU vendor and the cipher library (taken from a stable repository) are not included in the TD analysis since this code is not touched by the IMD developers under normal conditions.

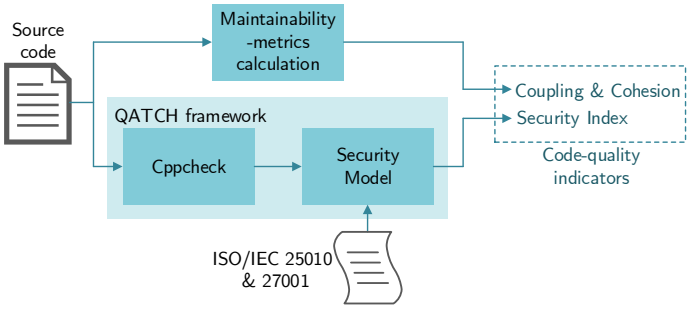


Figure 6.3: *Tool flow employed for code-quality measurement*

6.2.3 Software-quality measurement

By fixing the hardware platform, we guarantee fairness at the hardware level. In order to keep the comparison of the different software releases also fair and minimally dependent on our coding skills, we employ *static code analysis* in order to determine whether the code-quality (and vulnerability) levels were maintained throughout the various releases.

We, thus, employ the QATCH framework [143], which is based on a static-code analyzer called Cppcheck (see Figure 6.3). The analyzer is configured to detect security issues that reside in the source code. A *security model* aggregates the results produced by Cppcheck based on a set of international quality and security standards (i.e., ISO/IEC 25010 and ISO/IEC 27001) and produces a single score, the security index, which reflects the internal security level of the analyzed software. Moreover, the coupling and cohesion metrics generated by the maintainability-metrics calculator (see Section 6.1.1) are also used for measuring maintainability, extendability and understandability of the different releases.

6.2.4 Threat model

To understand all hardware and software design choices made and captured in the codebase record, it is imperative to establish a threat model for the documented IMDs. We assume a very pessimistic model that was introduced in Chapter 2 (Section 2.2.2). As a result, the IMD-security system has to satisfy certain security requirements, which were discussed in detail in Chapter 4 (Section 4.1).

6.3 Timeline of IMD design releases

The *time period* of the IMD-codebase record extends from 1997 to 2028 (see Table 6.1). Without loss of generality, the timeline up to 2020 is mostly based on historical data, which is cited in detail. The timeline beyond 2020 is fictitious, yet is generated by the *conservative* inclusion of security features commonly proposed in

current literature. Most post-2020 changes are security-related except for the use of multi-sensor recordings in next-generation neurostimulators to enable seizure prediction. This is used as an instance of a widely accepted medical design point in the future [78]. All in all, these design points are *not* the only choices available, but are taken as representatives of a general trend. It should be stressed that whether they will find their way in commercial IMDs or not does not affect the message of the chapter, which is a cautionary tale: By blindly incorporating ever-expanding medical/security provisions in future IMDs, the economic repercussions for manufacturers will be dire.

In order to construct the historical timeline, we opted for *yearly* code commits as we found this to be a realistic time resolution. The timeline (of past code releases) is based on the earliest reported date in literature regardless of the implant class. This is also important for quantifying the TD impact of each individual feature (at each code commit), which would have been lost with coarser time resolution. We should stress, however, that the timeline resolution does not impact in any way the TD analysis; only the right sequence of code commits is relevant. Still, we chose to include particular timestamps so as to correlate with the historical IMD development (see Table 6.1).

In what follows, we detail the IMD code releases. This timeline is very comprehensive, yet is necessary for clearly documenting our steps and for providing a strong experiment basis. The interested reader can skip the remainder of this section, proceed to the results discussed in Section 6.4, and return here for more details on the code releases. We denote the releases by $R_{<release\ \#>}$, summarized in Table 6.1.

$\mathbf{R_1}$: This code release implements the basic closed-loop medical functionality, as discussed in Section 6.2.1. A low-energy timer interrupt is used to wake up the MCU every $1/f_s$ seconds. The internal ADC is then used to sample the physiological signal. Upon processing the data to determine if the stimulus is needed or not, the MCU goes back to sleep.

$\mathbf{R_2}$: In this release, an RF-communication interface is added to the IMD for configuration updates. For the example applications, the filter-coefficient values and the threshold values of the detection algorithm can be read and/or configured via the wireless interface. Moreover, the treatment can also be turned on or off. We took an implantable-grade transceiver (TRX) [109] as an example. This transceiver communicates with the MCU via an SPI interface, in which the MCU acts as the master. Upon receiving the data, the TRX sends a GPIO interrupt to the MCU so that it can retrieve it from the TRX buffer. Hence, additional code is added to R_1 in order to enable this wireless interface. It also includes the decoding of user commands, based upon which the IMD performs the required actions. Moreover, the IMD application also formats the data to be sent in bytes in order to use one of the MCU USARTs as the SPI master and does the opposite for the received data.

The release date corresponds to the year when the Medical Implant Communication Service (MICS) was created by the FCC and a separate band was allotted for IMD communication [119]. Although RF-communication capability in IMDs existed long before MICS, this year marks the first year of standardized implant communication.

R_3: The IMD is now able to emit basic data logs, such as the recorded ECG/ECoG values. In addition to determining the treatment status, these logs can also be used for device diagnostics and troubleshooting purposes. Among the earliest IMDs to do this were the Medtronic Kappa 400 series pacemakers [101].

R_4: The application can now get the voltage level of the battery via its ADC and send it to the reader when asked by the user. Moreover, it also includes an audio-tone-based notification system, to mimic the ones that exist in the vintage Medtronic GEM III series pacemakers [100], which alerts the patient when the battery level is too low. In this system, the application periodically measures the voltage level (daily in our examples) and determines if it is below a certain threshold. In case of a low battery level, a small speaker is enabled for 10 seconds via one of the MCU GPIO pins.

R_5: This release introduces a watchdog timer as a safety mechanism. The timer resets the system to recover from a faulty condition, which could be due to a design bug or an external event that puts the MCU in an unknown state, making it unresponsive. As an example, an MCU that is stuck during electrical stimulation can cause serious complications on the patient's health. Such timers can be found in many MCUs including the MSP430 series [168].

R_6: In case of a software bug or a major functionality change, the IMD firmware has to be updated. Manual firmware updates imply surgically explanting the IMD, which is a risky and costly endeavor. Therefore, ideally the implant should be able to update its firmware wirelessly; i.e., *over the air (OTA)*. Based on our review of the past FDA advisories, we found that the earliest prescribed IMD-firmware update was reported in 2003 for a St. Jude Medical pacemaker (ADx pulse generator) launched in the same year [47, 157].

In release R_6, the firmware update is made possible using an *application bootloader*. In contrast to a *standalone bootloader*, which directly overwrites the existing application image in the instruction memory through a serial interface such as UART or SPI, the application-bootloader update is a two-stage process. The existing application first downloads the new image (via the transceiver) into an external flash or a vacant portion in the main (internal) flash[3]. It then calls the application bootloader to validate the new firmware image and copy it from the download space to the code space in the internal flash. The advantage of using an application bootloader, especially in a life-critical medical device, is that any errors introduced dur-

[3]The choice of flash for downloading the image does not have an impact on TD since the corresponding change in the source code is negligible.

ing the downloading stage do not negatively impact the running application. This is because the entire image is downloaded and its integrity verified before starting the actual update [154].

$\mathbf{R_7}$: This release implements more detailed data logs, which can be retrieved by the physician. These logs include the exact time stamps of certain events, e.g., epileptic seizures. This is made possible by using a real-time clock (RTC) module, which started appearing in some MSP430 parts (MSP430FG47x) around this time frame [169]. In this release, the user is also able to set the date and time of the device via the wireless interface.

$\mathbf{R_8}$: Due to the multiple reported vulnerabilities in IMD systems over the last decade or so and the strict measures taken by the FDA, we have finally started seeing standardized data-encryption implementations in these systems. For instance, the Azure pacemaker from Medtronic [103] implements NIST-standard encryption.

Release R_8 implements an ISO/IEC 9798-2-based, three-pass, mutual-authentication protocol, which is based on a pre-shared symmetric key between the reader and the implant. For data *confidentiality*, i.e., encrypting the reader commands and the IMD responses, the lightweight block-cipher SPECK is employed with block and key sizes of 64 and 128 bits, respectively. SPECK has been standardized in ISO/IEC 29167-22 as part of the RFID air interface standard (ISO/IEC 18000). For *authentication* and data *integrity*, a Cipher-based, Message-Authentication Code (CMAC) is employed, which generates a 32-bit MAC. Similarly, SPECK is used in *counter* mode to generate a fresh, 32-bit pseudo-random number (nonce) for replay protection. The interested reader can refer to [164] for a detailed description of the protocol and algorithms used.

It is important to note that we did not include the C-code implementation of SPECK in the TD analysis. This is because usually such code is taken from a stable repository and left untouched by the IMD developers.

$\mathbf{R_9}$: Based on past ethical-hacking efforts on IMD systems, DoS attacks have entered the fray as one of the easiest attacks to mount [31, 33, 62]. As discussed in Chapter 3, one of the effective ways of protecting against function DoS is – next to the main, medical MCU – to introduce a second MCU in the IMD for handling communication packets and security. Battery DoS can be prevented by initially operating this security MCU and the radio transceiver on the energy harvested from the incoming RF signal and allowing them to use the battery supply only after the external entity is authenticated. The steps involved are captured by the finite state machine (FSM) of the security MCU, as shown in Figure 6.4. The updated IMD design is, then, shown in Figure 6.5. The signal to switch the security MCU and TRX power supply comes from the security-MCU GPIO pin, as shown in the figure. The two MCUs are connected via the SPI interface in which the security MCU acts as the master.

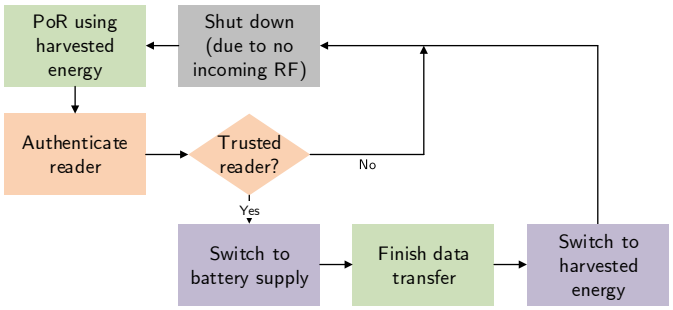


Figure 6.4: *State machine of secondary, security MCU (PoR: Power-on reset)*

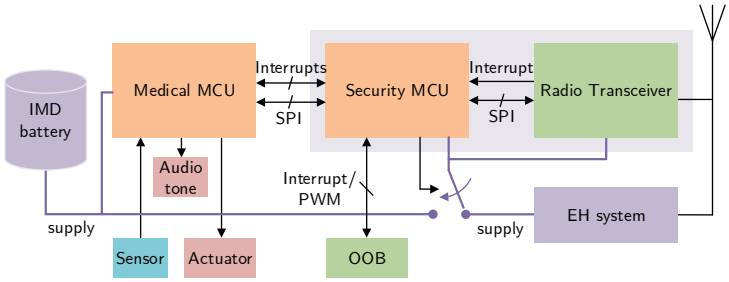


Figure 6.5: *System overview of final IMD design, including DoS resistance (R_9) and emergency access (R_{13})*

Various research works have advocated this dual-MCU approach in latest literature, [43, 62, 164, 186], which provides strong motivation for the industry to adopt in the future. As a result, this release onwards, we consider two separate C applications for TD analysis. It should be noted that the use of the watchdog timer from R_5 in the medical MCU also ensures that the medical treatment will be resumed in case communication is disrupted due to a disturbance in the wireless power transfer, which is a potential risk that R_9 introduces.

R_{10}: It is very much possible that a security primitive employed in an IMD becomes outdated after a certain time due to newly reported attacks on the primitive or due to the availability of better alternatives in terms of security, energy consumption and/or performance. To reflect this in our analysis, in release R_{10}, SPECK is replaced by the more secure AES-128. Many modern MCUs have a dedicated crypto peripheral that implements AES-128, among other primitives [153]. In this release as well, the security MCU uses its internal AES-128. It is important to note that even though a hardware implementation of the cipher is used, the required change will still be in software since the crypto peripheral sits within the MCU.

R_{11}: In the past, IMDs could only be accessed by the patient's physician. Modern IMDs, on the other hand, allow access to multiple users [103]. As a result,

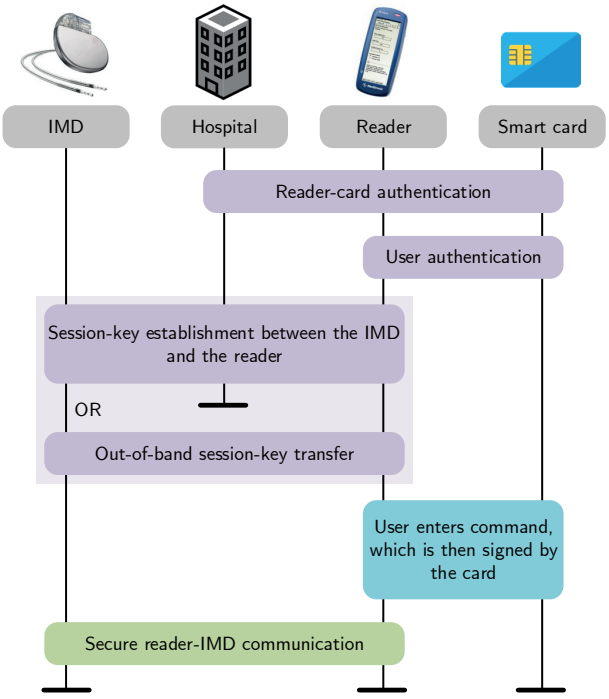


Figure 6.6: *Overview of the security protocol employed in* R_{11}-R_{13}

there is an increased possibility of medical mistakes, malpractice or even insider attacks. Therefore, *non-repudiation* is required to enforce user accountability (see Section 4.1.2). Moreover, the current landscape also requires *access control* so that a user is only able to send commands to the IMD according to her allowed privileges (see Section 4.1.5). Finally, the possibility of using multiple readers implies that the security based on pre-shared keys is not practical (see Section 4.1.7). Hence, *secure-key management* is also needed.

Similarly to the DoS discussion for R_9, we observe an increased focus in recent literature on providing the above services due to the nature of the emerging threats. Many of these works, such as [52, 116], propose the use of additional entities, i.e., a user smart card and a trusted third party. In R_{11}, one such protocol, IMDfence, which was proposed in Chapter 4, is implemented in order to provide the above security services. A brief overview of the protocol is shown in Figure 6.6. Non-repudiation is enabled by employing the signature of the command, which is signed by a personal smart card, and is sent to the IMD along with the command itself. Moreover, a hospital server is added to the overall system as a trusted third party in order to enable key management, access control and user authentication. Hence, this version requires the reader to be connected to Internet.

The IMD stores the signature so that it can be retrieved for dispute resolution in case the corresponding command results in the deterioration of patient's health. The signature must, therefore, be stored in a non-volatile memory, e.g., in the security-MCU flash memory, to protect against MCU resets.

$\mathbf{R_{12}}$: Accurate prediction of epileptic seizures is an open topic in the neuroscience research. Neurostimulators are ideal candidates to enable such a treatment since they are already used for seizure suppression. One of the prominent research directions is to add multiple sensors to the implant in order to improve the prediction accuracy [78]. In order to capture any demanding future medical enhancement, R_{12} mimics the above scenario in which the closed-loop IMD system is based on multi-sensor inputs.

In this version, the MCU ADC is used in *scan* mode to sample multiple sources. In order to process these samples, a separate filtering operation per each added input source is required. Since only one MCU is employed for the medical application, these executions have to be performed sequentially, which increases the active-vs-sleep duty cycle of the MCU.

$\mathbf{R_{13}}$: Another important security feature that is touted in modern literature is *emergency access* [43, 52, 132], which does not exist in IMDs at present (see Section 4.1.3). Since the paramedic reader and IMD do not share a secret, the protocol from R_{11} can still be used. However, it cannot work in the absence of an Internet connection, which can help establish trust remotely.

Release R_{13} solves this problem by using an out-of-band (OOB) channel, such as galvanic coupling, ultrasound communication etc. Similarly to the SecureEcho approach (Chapter 5), this channel is used to pair the reader and the IMD by transferring a fresh symmetric key to enable secure RF communication (see Figure 6.6). One approach is to employ ASK modulation along with PWM encoding of bits in the reader-IMD OOB channel. Bits 0 and 1 can be differentiated by choosing different PWM duty cycles for each. The security MCU wakes up (via an interrupt) on the rising edge of every received bit. It then records the value on the same GPIO pin after a certain time period (with the help of a timer) in order to determine the bit level (1 or 0). The OOB data rate does not have to be high since the volume of data to be transferred, i.e., the session key, identifiers and nonces, is very low. The system architecture of the final IMD design is shown in Figure 6.5.

6.4 Experimental results

In this section, we present the results of the TD analysis, which tries to capture the repercussions of introducing security at a certain point in the IMD-development timeline, and its interplay with the medical application.

Table 6.2: *Summary of IMD resource usage (R_{13} (2028))*

	Lifetime* (years)		Delay** (ms)	Prog.-memory
	Neuro	Cardiac		footprint (kB)
Without Security	8.7	14.6	20.3	27.5
With Security	7.1	11.7	85.9	53.1§

* For a typical IMD battery size of 9.5 Ah.
** It includes security-processing and TRX (SPI data handling and RF transmission) delays pertaining to a communication session in which 256 bytes of filter coefficients are read from the IMD.
§ It includes the program-memory footprint of *both* MCUs.

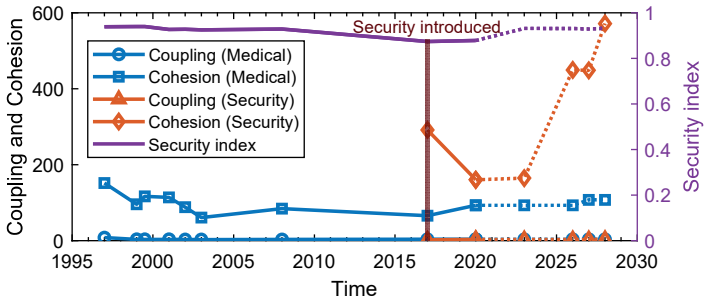


Figure 6.7: *Overview of the code quality across all the IMD releases for both the medical and the security codebases.*

6.4.1 Checking technical feasibility

We first briefly perform an IMD-autonomy and -performance analysis of the final design (R_{13} (2028)) in order to ensure that our experimental code does not introduce prohibitive energy, processing and area overheads. For this analysis, we reuse the MCU and transceiver operating conditions and the pacemaker energy consumption from Chapter 3 (see Table 3.3). The neurostimulator energy consumption during stimulation is borrowed from an actual seizure-suppression system [112]: under worst-case conditions, the stimulation current, pulse width, pulse frequency and burst duration are assumed to be 12 mA, 1 ms, 333 Hz and 10 seconds, respectively, with an average of 4.3 seizures per day [10].

The results are summarized in Table 6.2, which shows that the IMD resource usage is within acceptable limits: the calculated battery lifetime is sufficiently long, the security-processing and communication delays are negligible, and the program-memory footprints are small. These results also show that security does not significantly impact IMD autonomy.

6.4.2 Checking code quality

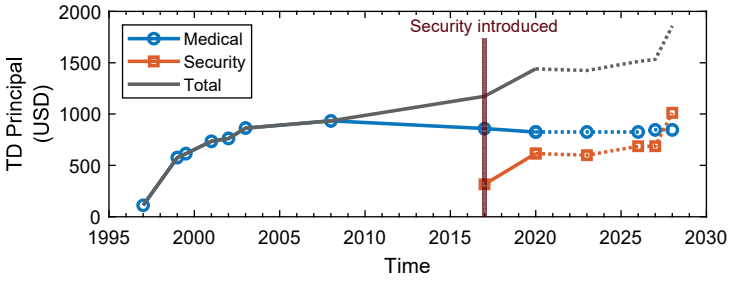
Figure 6.7 provides the results of the static code analysis introduced in Section 6.2.3. The **security index** is high across all the releases, which means that the changes that were made throughout these versions do not introduce new code vulnerabilities. Moreover, **coupling** and **cohesion** values stay fairly constant across all the releases, indicating that a consistent code quality was kept throughout the analyzed timeline.

6.4.3 Technical-debt analysis

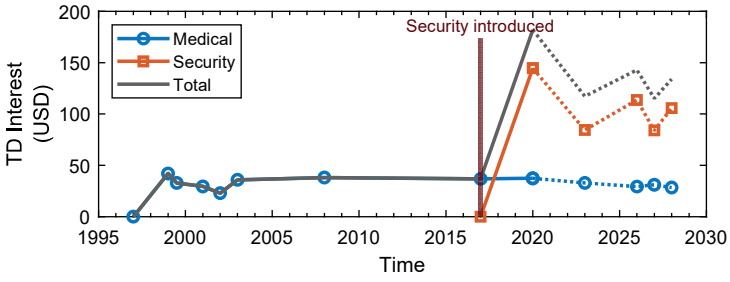
The TD principal, interest and accumulated TD of the IMD application-code releases are summarized in Figure 6.8. As mentioned in Section 6.1.1, an hourly rate of USD 45.81 is used for the TD calculations. It should be clarified that the TD costs correspond to *additional* repayment and maintenance activities and they do not represent the total development costs, as previously illustrated in Figure 6.1; these activities correspond to coding and verification efforts only.

We notice a steep increase in the *TD principal* at R_2 (1999), since therein is implemented the serial interface with the transceiver and the command decoding. This, interestingly, indicates that the wireless-interface-related code forms the major component of the application instead of the medical functionality. Moreover, because of this large increase, the next release (R_3 (1999)) causes a relatively steeper rise in the *TD interest* due to the increased maintenance effort. The decline in the TD principal of the medical application at R_8 (2017) and R_9 (2020) is because the communication-related processing in the medical code was moved to the security code. However, this reduction does not match the corresponding rise in the security-code principal during the same period due to the security-protocol implementation. What is more, two serial interfaces – one to the transceiver and one to the medical-application MCU – are added in R_9 (2020). It is important to note that the observed rise in the security-code TD principal does not include the cipher library in the analysis, as mentioned in the R_8 description. As a result, we do not see any noticeable change in the principal costs when replacing the cipher (i.e., SPECK with AES-128) since only the associated wrapper functions required change (R_{10} (2023)).

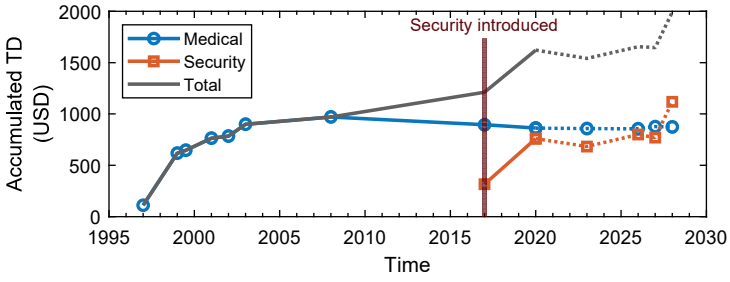
Having explained the reasons behind the morphology of the TD curves, let us now take a step back and assess the information they offer us. We can see that the total **TD principal** (grey line) reaches a maximum cost of just under USD 1,900, whereas the total **TD interest** for individual releases stays below USD 190. From either of the two curves, it can be deduced that *the security code is indeed more costly to extend and maintain than the medical code.* The interest, especially, is at least double for the security component. As a result, the **accumulated TD** (Figure 6.8c) is mostly driven by that component. By inspecting the total trend line, it is also interesting to notice that *security-driven code changes will eventually overtake medical-driven ones in the future.* Yet, we should pay attention to the actual cost these changes incur, as



(a) TD Principal



(b) TD Interest

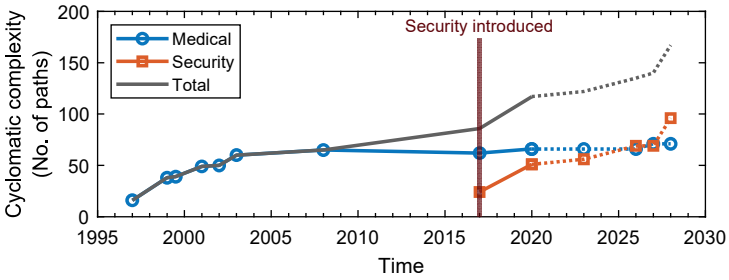


(c) Accumulated TD

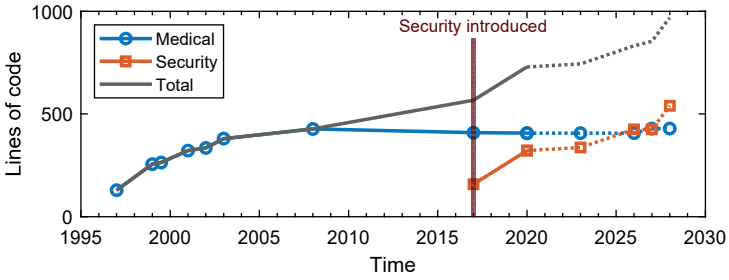
Figure 6.8: *Overview of total and per-IMD codebase (medical, security) TD metrics. Solid lines indicate existing, documented IMD-code features, while dotted lines show future, projected features. Costs are calculated based on the default SonarQube hourly rate ($45.81).*

predicted by the analysis tools: Accumulated TD reveals that additional IMD-code repayment and maintenance costs are limited to only a few thousand dollars in the near future but those can *drastically deteriorate* for IMD manufacturers in the longer term, if left unchecked.

In Figures 6.9a and 6.9b, two of the components for calculating the TD interest have also been plotted: cyclomatic complexity and lines of code, so as to offer more insights on our application behavior. We see that the **cyclomatic complexity (CC)**



(a) Cyclomatic complexity



(b) Lines of code

Figure 6.9: *Overview of CC and LOC for the security and medical codebases.*

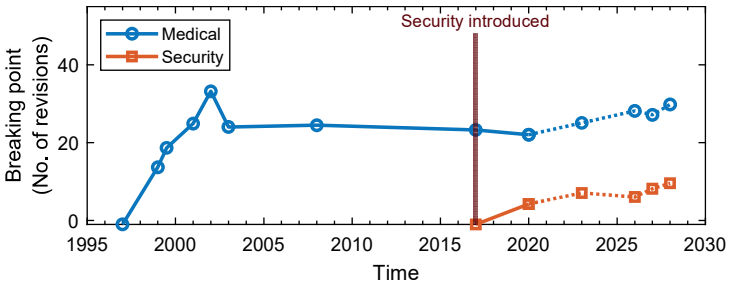


Figure 6.10: *Overview of the breaking point for the security and medical codebases.*

of the security code increases at a faster rate than that of the medical code due to the type of complexity involved in the respective applications: Even a minimal change in the security protocol results in a significant increase in the corresponding FSM complexity (see Figure 6.4). For instance, numerous fallbacks are added so that the IMD FSM returns to a stable state in case the communication is disrupted midway. Besides, a significant portion of the IMD code is composed of control-flow statements. As a result, the **Lines-of-Code (LOC)** curve is very similar to the cyclomatic-complexity curve. Note that the LOC values seem to be relatively low. This is because the low-level peripheral-support library from the MCU vendor and

the cipher library are not included in the TD analysis since they are not modified by the IMD developers.

The higher CC of the security code is, finally, also reflected in the **breaking-point** curves plotted in Figure 6.10; The potential breaking point of the security code, at a given point in time, can be reached significantly earlier than that of the medical code. This observation indicates that the security code should be developed after careful planning. It also tells us that the medical code is easier to maintain since its breaking-point curve is always higher than the security curve.

6.4.4 Discussion

The work in this chapter was carried out in order to answer the critical question whether adding security to modern IMDs is an economically viable and sustainable venture for IMD manufacturers (and other stakeholders). In the face of a rising number of cybersecurity attacks, this question becomes very relevant and time-sensitive.

The analysis has revealed that *adding* security code to an IMD medical-only codebase is going to be more difficult (in effort, and thus in cost) than adding new medical code, as the TD-principal estimations reveal (Figure 6.8). It will also be more difficult to *maintain* the security code compared to the medical one. These difficulties translate to higher development costs, which stem from the fact that the security codebase is generally more complex, more volatile and can deteriorate or break more easily. Fortunately, such software costs are rather low and can be shouldered by manufacturers.

Our analysis has necessarily relied on a synthetically constructed codebase; however, should our TD projections be accurate, the *security-driven TD can become critical in the future*. This finding is worrisome given that the security provisions of future IMD systems will grow to encompass also IMD readers (see Figure 6.6) and even remote IMD-company servers, each extra component introducing its own security codebase. In this context, security-driven TD is expected to rise even more steeply. Therefore, unlike the medical codebase (which in many cases remains practically unchanged across IMD generations), the security codebase has to be frequently refactored for the overall TD to remain in check.

The above findings lead to the main conclusion that present-day IMDs can be financially tractable with (perhaps necessarily) "quick and dirty" security solutions but this modus operandi has to transition soon to a more structured security-development approach so as to keep development costs under control and, thus, the viability of future IMD systems high for IMD manufacturers, insurance companies, healthcare systems and, eventually, patients themselves.

6.5 Related work

Technical debt is a widely used concept in software engineering. However, its use in improving software security has not been explored in detail [127]. Siavvas et

al. [142] investigated the potential relationship between TD and software security, based on a relatively large repository of popular open-source software applications. Their preliminary findings suggest that TD, apart from quality issues, may potentially indicate the existence of security-vulnerability issues in a software. Similarly, TD has only recently been considered in energy-efficient software design for embedded systems. In this context, most of the emphasis has been on studying the impact of code refactoring on the energy consumption [115, 120]. Example domains include mobile applications [115] and vehicular technology [41]. However, the applicability of TD in IMD systems and other related domains, such as wireless body area networks (WBANs), has not been explored.

Fu [54] was the first to bring up TD in the context of medical devices. He pointed out that hackers can be regarded as the *messengers* of cybersecurity TD because they uncover the implications of the flaws that exist due to poor design choices. However, the discussion does not go in depth regarding the repercussions of amassed TD in view of securing future IMDs.

All in all, our work departs from the previous works by performing a comprehensive TD analysis to study the impact of adding security to modern IMD systems.

6.6 Summary

In the recent past, there has been a significant ramp-up in IMD ethical-hacking activities. The regulatory bodies worldwide are also increasing their pressure on the IMD manufacturers to improve the security of these devices. In this chapter, we embarked on a methodology to quantitatively analyze the cost of adding security in the existing devices from the perspective of embedded-software technical debt (TD). This is the first time that TD, which is a relatively new concept, has been used to analyze this class of embedded systems. By necessarily relying on a synthetically constructed IMD codebase, we found that security software, on one hand, is costlier to develop and maintain than the preexistent, purely medical software in IMDs but overall costs are insignificant in the short term. On the other hand, the higher complexity and volatility of the security codebase is projected not only to dominate future costs but also to disrupt the economic viability of IMD products in the next decade, if the IMD-software TD growth is left unchecked.

CHAPTER 7

7

Conclusions

7.1 Summary

The main goal behind this thesis has been to rethink the way we approach IMD security, and to debunk the prevalent myths in this domain. Specifically, the thesis focuses on addressing the most pertinent security concerns by proposing secure reader-IMD-communication protocols that provide a robust security portfolio as well as assessing the economic feasibility of adding security to IMD systems.

In order to lay a proper foundation, we first analyzed the IMD-security landscape using a systematic threat-modeling approach in Chapter 2. This attack-tree-based approach offers a highly structured picture of the strengths and weaknesses of IMD systems. Taking three IMD secure-communication protocols found in literature as a case study, we highlighted the security services requiring most focus in this domain, i.e., IMD availability (or in other words, protection against DoS attacks) and non-repudiation. Using our threat analysis, we also discovered certain limitations in the above protocols, and subsequently provided recommendations to strengthen them. This work, thus, paves the way for building more robust and secure protocols and protection mechanisms for these systems.

In Chapter 3, we focused on IMD availability, which was earmarked as an area of concern in the above attack-tree-based analysis. We looked into battery DoS, which is among the easiest and most effective attacks to mount from an attacker's perspective. We provided an extensive review of works from literature that protect against such attacks. After formulating an extensive list of design considerations, these works were analyzed and their shortcomings highlighted. Subsequently, we provided recommendations towards realistic zero-power-defense (ZPD) implementations, including the concept of adaptive ZPD, which facilitates remote monitoring, and the standalone ZPD module, which reduces the IMD time to market.

In Chapter 4, we *extended* the necessary security requirements from Chapter 2. These include availability, non-repudiation, emergency access, access control, entity authentication, remote monitoring and system scalability. These requirements formed the basis of a novel security protocol for IMD ecosystems termed IMDfence. We performed the automated validation of this protocol's handshake-specific requirements using the widely-accepted AVISPA tool. We also showed that IMDfence does not introduce any noticeable overheads in the implant, and it has the ability to support ZPD against battery-DoS attacks. Our evaluation demonstrated that IMDfence achieves the above security requirements at a mere 6.57% increase in total IMD consumption, which is minimal in the context of an IMD lifespan, and less than a 14-ms and 9-kB increase in system delay and memory footprint, respectively.

In Chapter 5, we discussed in detail the OOB-pairing approach introduced in Chapter 4, which enables offline or emergency access. We presented SecureEcho, an ultrasound-based, device-pairing scheme for reader-IMD systems that inherently provides protection against battery DoS. We also performed a first ever security evaluation of the ultrasound channel. Our results indicate that, when using trans-

ducers that are sensitive to frequencies ≥ 1 MHz, it becomes infeasible for the attacker to eavesdrop or insert messages even from a range of a few millimeters. Furthermore, we demonstrated a proof-of-concept implementation of the passive circuit that enables the pairing process and ZPD. Through our physical experiments, we showed that the TX energy transferred through the medium is small enough to fall within the budget of a battery-powered portable reader. Moreover, the acoustic power transferred through the medium is comfortably within the FDA safety limits. Thereafter, we discussed how SecureEcho outperforms the traditional harvesting-based ZPD in terms of satisfying frequency-band and medical-safety constraints, operating range, emergency access, design suitability and dependability.

After focusing on the IMD-protection mechanisms and security protocols, we took the discussion to a different direction in Chapter 6. We systematically analyzed the cost of adding security to existing IMDs from the perspective of technical debt (TD). This was achieved by first constructing a synthetic historical record of IMD design changes, captured as software releases over time. Both medical and security aspects were targeted in these releases. Our evaluation revealed that security software is costlier to develop and maintain than the purely medical software in IMDs. Fortunately, such costs are insignificant in the short term. However, due to the highly complex nature of security software, these costs are projected to become critical if TD growth is left unchecked. These findings indicate that the modus operandi of addressing security issues in a "quick and dirty" way has to change to a more structured approach.

7.2 Scientific contributions

The main scientific contributions of this thesis are recapped as follows:

1. **Attack-tree-based threat modeling of IMDs:** We established attack trees for IMDs, which, to the best of our knowledge, is the first work formulated for these devices. This work makes possible the creation of a constantly expanded reference point by and for the whole IMD community.

2. **Systematization of knowledge of ZPD mechanisms:** We consolidated a comprehensive list of ZPD design considerations for the specific domain of IMDs. We subsequently performed a survey of existing systems and highlighted their limitations based on the above considerations.

3. **Practical recommendations for ZPD mechanisms:** We provided recommendations for developing comprehensive protection of IMDs against battery-DoS attacks. We proposed the concept of adaptive ZPD, which allows the IMD to communicate with a bedside reader (i.e., over a long range) while still protecting against battery-DoS attacks. Moreover, we proposed the notion of a

standalone ZPD module, which significantly reduces the IMD-certification effort.

4. **A novel security protocol for IMD ecosystems:** We proposed a comprehensive secure-communication protocol, IMDfence, which addresses crucial, yet previously ignored security requirements, i.e., non-repudiation, remote monitoring and system scalability. Moreover, we showed that the protocol allows emergency access that results in the graceful degradation of the supported security services without compromising device security and patient safety.

5. **An extensive evaluation of our proposed security protocol:** We performed a rigorous validation of IMDfence using a formal tool (AVISPA). Moreover, we extensively evaluated the protocol paying special attention to ZPD, and its impact on the IMD lifetime and performance.

6. **A novel reader-IMD device-pairing scheme:** We proposed SecureEcho, a lightweight device-pairing scheme that is based on ultrasound communication. SecureEcho consists of a handshake protocol and a system architecture that only enables RF communication after an external entity is authenticated. We showed that this configuration achieves ZPD without requiring any RF-energy harvesting, which significantly reduces the IMD design complexity.

7. **A comprehensive security analysis of ultrasound as an OOB channel:** We performed an extensive security analysis of the ultrasound communication channel employed in SecureEcho using state-of-the-art acoustic wavefield simulations. Our results conclusively showed that the attacker would not be able to successfully eavesdrop or insert messages, or launch battery-DoS attacks.

8. **Implementation and evaluation of our device-pairing scheme:** We demonstrated a proof-of-concept implementation and validation of the SecureEcho approach. We showed that it can be easily incorporated into reader-IMD systems and that it complies with the FDA safety limits.

9. **A synthetic historical record of IMD design changes:** We carefully constructed a synthetic IMD-codebase record of medical- and security-functionality changes, captured as software releases over time. The record is freely available to download with the aim of putting it under public scrutiny and helping IMD stakeholders.

10. **Determination of the economic viability of adding security to IMDs:** We performed a systematic analysis of security-related software costs in IMDs using the concept of technical debt. Our results showed that security-related

changes amass more TD than medical ones but the overall costs are manageable in the short term. However, if the TD growth is not kept in check, it can disrupt the economic viability of adding security to IMDs.

11. **Determination of the technical feasibility of adding security to IMDs:** We provided a short technical-feasibility study of inserting mainstream security mechanisms in commercial IMDs. We performed this analysis taking into consideration the two most common IMD classes in the market, i.e., cardiac implants and neurostimulators. Our results revealed that the impact of adding security on IMD battery lifetime, performance and program memory is within acceptable limits.

7.3 Future directions

As is the case in most PhD theses, we ended up with some unanswered questions and untraversed challenges at the end of this doctoral-research journey. Due to the limited time and resource constraints, these unexplored scientific problems are left for future work and are discussed next.

Attack trees

Besides performing a threat analysis of IMD systems, the intention behind constructing attack trees was also to ultimately develop an open-access resource where current IMD-security-research efforts can reflect upon and also contribute to. Such a threat-landscape directory or a database, which is formulated in an attack-tree format, would make it straightforward to automate the likelihood calculation of each threat. Moreover, enabling medical experts to access such a database could help in extending the attack trees with operational security aspects based on past experiences from medical practice.

Zero-power defense

The key recommendations towards practical harvesting-based ZPD implementations include, among others, the concepts of *adaptive ZPD* and the *standalone ZPD module*. While we have shown that these recommendations are conceptually possible, we do realize that this discussion was at a very high level. Hence, prototyping is required to actually be sure if these notions are realizable. In the case of adaptive ZPD, which facilitates bedside-base-station operation, one approach could be to implement a finite state machine within the security processor/MCU that causes the IMD to switch between RFPT (for base-station/IMD communication) and IPT (doctor-programmer/IMD communication). Regarding the standalone ZPD module, which reduces the IMD-certification effort, one of the constraints is that there should be enough space inside the IMD casing for the placement of this module. Since our discussion was focused around typical IMDs available commercially, which have

sufficient vacant space, this issue was not investigated in detail. However, a prototype can provide new insights on how far we can go with the miniaturization so as to gauge this concept's relevancy in next-generation devices.

When it comes to our non-harvesting-based ZPD approach, SecureEcho, the physical implementation was done at a proof-of-concept level. Enhancing it further to form a refined prototype can provide us with new insights. Recall that SecureEcho could be susceptible to side-channel attacks if electromagnetic shielding over the ultrasound circuitry is not in place. Since such a shielding was not incorporated in our proof-of-concept implementation, it would be interesting to see whether including it in a future prototype uncovers any hidden challenges. Moreover, the use of smallest possible form factor components can guide us about the level of miniaturization that can be achieved for our approach.

Security protocol for IMD ecosystems

In IMDfence, we introduced the concept of reader-card-authentication lifetime (T), which allows a paramedic to access an IMD in an offline setting without requiring reader-card authentication. A prolonged offline operation enabled by a very large T may result in the reader's and/or IMD's firmwares becoming outdated. On the other hand, a very small value hinders legitimate access, i.e., availability. Therefore, T should be assigned an appropriate value (within maximum and minimum limits) taking into consideration a variety of factors. For instance, a sensible option would be to assign a lower T value for urban areas compared to rural environments. This is because the probability of having stable Internet connectivity is higher when the patient is living in an urban area compared to a rural region. Additionally, it stands to reason that the likelihood of attacker presence ought to be higher in an urban environment. Besides patient locality, reader-IMD usage patterns should also be taken into consideration. These depend on the patient condition and IMD type (e.g., cardiac implants, neurostimulators etc.). The IMDs requiring frequent reader access should be granted a larger T value. It would be interesting to perform further investigation on how to better optimize T while balancing usability and availability.

Body-coupled communication

Recall that in Chapter 5, we hypothesized that SecureEcho approach could also work if ultrasound communication is replaced by galvanic coupling, though we prefer the former. This is because ultrasound transducers offer highly directional and extremely-short-range communication depending on the frequency of operation and transducer width, which was also corroborated by our security analysis. This makes MHz-range ultrasound ideal for secure key transport. However, it would be interesting to perform a similar in-depth security analysis for galvanic coupling and pinpoint the acceptable ranges of operation. This would allow for subsequent prototyping and, as a result, a comprehensive comparison with ultrasound.

Another OOB channel of interest is inductive coupling[1] (IC), such as NFC. It was not investigated in this thesis because it is suspected to be less localized and, thus, more vulnerable to eavesdropping compared to ultrasound and galvanic coupling [173]. It is interesting to note that Medtronic [105] and Biotronik [18] use IC as the initial (unencrypted) short-range communication in order to activate the long-range RF telemetry for regular reader-IMD communication. This implies that their threat models consider IC to be secure from eavesdropping and message-insertion attacks. However, there is no such study that definitely indicates it to be the case. If IC is proven to be insecure, the security of the overall system will collapse. Some researchers reject the above threat model [94] and opt for more conservative ones instead. The reason provided is that it could be possible for an attacker to come really close to the patient in crowded areas (e.g., public transport during rush hours) and activate RF telemetry via IC. In order to definitely answer the concerns about this threat model, an elaborate security analysis of IC is required, which takes into account the attacker transceiver/antenna gain, distance, environmental conditions etc.

Many next-generation neuromodulation systems from literature have proposed the use of mm-sized neural implants for both the Central (CNS) and Peripheral Nervous Systems (PNS). These implantable systems consist of a multitude of mm-sized nodes with the aim of achieving high-spatial-resolution neural recording and stimulation [140]. For such systems, however, the security research is non-existent. This is because these implants are highly resource constrained and, as a result, executing cryptographic primitives on them looks infeasible for the foreseeable future. In Chapter 5, we discussed that ultrasound is being touted as an in-body communication channel between these implants [140, 185]. This is because of the smaller size of ultrasound transceivers compared to electromagnetic ones, which is ideal for scaling-down of IMDs, and the much relaxed medical-safety constraints. With this in mind, we can utilize the results from the SecureEcho security analysis (i.e., employ MHz-range transducers) in order to inherently secure the in-body communication without employing any cryptographic computations.

Technical-debt analysis

In Chapter 6, we claimed that *hardware* technical debt, would be mostly irrelevant for IMDs. This is because hardware changes never occur within a given IMD's lifetime (since it is implanted) and occur rarely within a given product line. As a result, modern IMDs can be considered as software-driven devices, which implies that hardware changes incur virtually no TD. Besides, these changes can be captured via their repercussions in the respective software, which changes far more frequently by comparison. To the best of our knowledge, there is no mature methodology that

[1] Note that in Chapter 3, inductive coupling was only discussed in its capacity as a wireless-power-transfer channel.

accurately estimates hardware TD. However, if such a methodology does become available, it would be still interesting to utilize it in IMD systems in order to get a new perspective on answering the IMD-economics question.

In the same chapter, we highlighted the four obstacles in getting actual IMD-application codebases from manufacturers in order to perform an *ideal* TD analysis. These obstacles resulted in employing a *synthetic* codebase, which was created based on publicly available clinician's manuals (from multiple manufacturers), news articles, data sheets, and so on. In order to further safe-guard our confidence in the codebase representability we employed auxiliary metrics and also made the codebase publicly available. However, it would still be highly valuable to perform the TD analysis on an actual codebase, assuming a manufacturer is willing to release its software. Furthermore, if this manufacturer keeps a comprehensive log of developer activity, it would be interesting to see if the *actual* maintenance effort can be correlated with the calculated TD interest.

A

Appendix

Bibliography

[1] Abbott. (2018) Prodigy MRI^{TM} Chronic Pain System. [Online]. Available: https://www.sjmglobal.com/en-int/patients/chronic-pain/our-neurostimulation-systems/prodigy-mri-chronic-pain-system

[2] Abbott Laboratories and The Chertoff Group, "Why Medical Device Manufacturers Must Lead on Cybersecurity in an Increasingly Connected Healthcare System," 2017, White Paper.

[3] B. Adida, M. Bond, J. Clulow, A. Lin, S. Murdoch, R. Anderson, and R. Rivest, "Phish and chips," in *International Workshop on Security Protocols*. Springer, 2006, pp. 40–48.

[4] R. AlTawy and A. M. Youssef, "Security tradeoffs in cyber physical systems: A case study survey on implantable medical devices," *IEEE Access*, vol. 4, pp. 959–979, 2016.

[5] A. Ampatzoglou, A. Ampatzoglou, P. Avgeriou, and A. Chatzigeorgiou, "Establishing a framework for managing interest in technical debt," in *5th International Symposium on Business Modeling and Software Design, BMSD*, 2015.

[6] R. Anderson, *Security Engineering*. Wiley, 2008.

[7] R. J. Anderson, "Liability and computer security: Nine principles," in *European Symposium on Research in Computer Security*. Springer, 1994, pp. 231–245.

[8] E.-M. Arvanitou, A. Ampatzoglou, A. Chatzigeorgiou, and P. Avgeriou, "Software metrics fluctuation: a property for assisting the metric selection process," *Information and Software Technology*, vol. 72, pp. 110–124, 2016.

[9] J.-P. Aumasson and A. Vennard, "Cryptography in industrial embedded systems: our experience of needs and constraints," in *NIST Lightweight Cryptography Workshop 2019*. NIST, 2019.

[10] M. Balish, P. S. Albert, and W. H. Theodore, "Seizure frequency in intractable partial epilepsy: a statistical analysis," *Epilepsia*, vol. 32, no. 5, pp. 642–649, 1991.

[11] J. Bansiya and C. G. Davis, "A hierarchical model for object-oriented design quality assessment," *IEEE Transactions on software engineering*, vol. 28, no. 1, pp. 4–17, 2002.

[12] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3056–3076, 2012.

[13] S. Barrett and D. Pack, *Microcontroller Programming and Interfacing TI MSP430*, ser. Synthesis Lectures on Digital Circuits and Systems. Morgan & Claypool Publishers, 2011, no. pt. 1.

[14] H. Basaeri, D. B. Christensen, and S. Roundy, "A review of acoustic power transfer for bio-medical implants," *Smart Materials and Structures*, vol. 25, no. 12, p. 123001, 2016.

[15] T. Belkhouja, X. Du, A. Mohamed, A. K. Al-Ali, and M. Guizani, "Symmetric Encryption Relying on Chaotic Henon System for Secure Hardware-Friendly Wireless Communication of Implantable Medical Systems," *Journal of Sensor and Actuator Networks*, vol. 7, no. 2, p. 21, 2018.

[16] ——, "Biometric-based authentication scheme for implantable medical devices during emergency situations," *Future Generation Computer Systems*, vol. 98, pp. 109–119, 2019.

[17] K. Bhargavan and G. Leurent, "On the practical (in-) security of 64-bit block ciphers: Collision attacks on HTTP over TLS and OpenVPN," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 456–467.

[18] Biotronik, *BIOTRONIK Statement on the Medical Advisory and Safety Communication Regarding Medtronic's Conexus Radio Frequency Telemetry Protocol*, 2019.

[19] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "ProVerif 2.02 pl1: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial," 2020.

[20] V. Blue. (2017) Turns out, pacemaker security is terrifying. [Online]. Available: https://www.engadget.com/2017-04-21-pacemaker-security-is-terrifying.html

[21] Bluetooth SIG. (2020) View previously qualified designs and declared products. [Online]. Available: https://launchstudio.bluetooth.com/Listings/Search

[22] M. Bon. (2016) A Basic Introduction to BLE Security. [Online]. Available: https://www.digikey.com/eewiki/display/Wireless/A+Basic+Introduction+to+BLE+Security

[23] L. Bu, M. G. Karpovsky, and M. A. Kinsy, "Bulwark: Securing implantable medical devices communication channels," *Computers & Security*, vol. 86, pp. 498–511, 2019.

[24] Businesswire. (2015) Silicon Labs Secures IoT Nodes with New EFM32 Jade and Pearl Gecko Microcontrollers. [Online]. Available: https://www.businesswire.com/news/home/20151214005228/en/Silicon-Labs-Secures-IoT-Nodes-New-EFM32

[25] C. Camara, P. Peris-Lopez, J. M. De Fuentes, and S. Marchal, "Access control for implantable medical devices," *IEEE Transactions on Emerging Topics in Computing*, 2020.

[26] C. Camara, P. Peris-Lopez, and J. E. Tapiador, "Security and privacy issues in implantable medical devices: A comprehensive survey," *Journal of biomedical informatics*, vol. 55, pp. 272–289, 2015.

[27] S.-Y. Chang, S. L. S. Kumar, B. A. N. Tran, S. Viswanathan, Y. Park, and Y.-C. Hu, "Power-positive networking using wireless charging: protecting energy against battery exhaustion attacks," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 2017, pp. 52–57.

[28] A. Chatzigeorgiou, A. Ampatzoglou, A. Ampatzoglou, and T. Amanatidis, "Estimating the breaking point for technical debt," in *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*. IEEE, 2015, pp. 53–56.

[29] H. Chi, L. Wu, X. Du, Q. Zeng, and P. Ratazzi, "e-SAFE: Secure, Efficient and Forensics-Enabled Access to Implantable Medical Devices," in *2018 IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1–9.

[30] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object oriented design," *IEEE Transactions on software engineering*, vol. 20, no. 6, pp. 476–493, 1994.

[31] CISA. (2017) ICS Advisory (ICSMA-17-241-01). [Online]. Available: https://www.us-cert.gov/ics/advisories/ICSMA-17-241-01

[32] ——. (2018) ICS Advisory (ICSMA-18-179-01). [Online]. Available: https://www.us-cert.gov/ics/advisories/ICSMA-18-179-01

[33] ——. (2020) ICS Medical Advisory (ICSMA-19-080-01). [Online]. Available: https://www.us-cert.gov/ics/advisories/ICSMA-19-080-01

[34] A. Costanzo, M. Dionigi, D. Masotti, M. Mongiardo, G. Monti, L. Tarricone, and R. Sorrentino, "Electromagnetic energy harvesting and wireless power transmission: A unified approach," *Proceedings of the IEEE*, vol. 102, no. 11, pp. 1692–1711, 2014.

[35] C. J. Cremers, "The Scyther Tool: Verification, falsification, and analysis of security protocols," in *International conference on computer aided verification*. Springer, 2008, pp. 414–418.

[36] W. Cunningham, "The WyCash portfolio management system," *ACM SIGPLAN OOPS Messenger*, vol. 4, no. 2, pp. 29–30, 1992.

[37] Cypress Semiconductor Corporation. (2017) Energy Calculation for Energy Harvesting with S6AE101A, S6AE102A, and S6AE103A. [Online]. Available: http://www.cypress.com/file/234931/download

[38] T. Denning, K. Fu, and T. Kohno, "Absence makes the heart grow fonder: New directions for implantable medical device security." in *HotSec*, 2008.

[39] M. Deterre, "Toward an energy harvester for leadless pacemakers," Theses, Université Paris Sud - Paris XI, Jul. 2013. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00868838

[40] Eaglepicher Technologies. (2018) Medical Power. [Online]. Available: https://www.eaglepicher.com/markets/medical-power/

[41] U. Eliasson, A. Martini, R. Kaufmann, and S. Odeh, "Identifying and visualizing architectural debt and its efficiency interest in the automotive domain: A case study," in *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*. IEEE, 2015, pp. 33–40.

[42] N. Ellouze, M. Allouche, H. Ben Ahmed, S. Rekhis, and N. Boudriga, "Securing implantable cardiac medical devices: Use of radio frequency energy harvesting," in *Proceedings of the 3rd international workshop on Trustworthy embedded devices*. ACM, 2013, pp. 35–42.

[43] N. Ellouze, S. Rekhis, N. Boudriga, and M. Allouche, "Powerless security for cardiac implantable medical devices: Use of wireless identification and sensing platform," *Journal of Network and Computer Applications*, vol. 107, pp. 1–21, 2018.

[44] Emergo. (2017) Compare the time, cost and complexity of getting regulatory approval for medical devices. [Online]. Available: https://www.emergobyul.com/resources/worldwide/global-regulatory-comparison-tool

[45] S. Evanczuk. (2014) Capacitor Characteristics Impact Energy Harvesting Efficiency. [Online]. Available: https://www.digikey.nl/en/articles/techzone/2014/nov/capacitor-characteristics-impact-energy-harvesting-efficiency

[46] FCC, "Medical Device Radio Communications Service," *Title 47*, vol. Chapter I, Subchapter D, Part 95, Subpart I, 2018.

[47] FDA, "ADX Pulse Generator Firmware Anomaly Correction," *Devices@FDA*, 2003.

[48] ——. (2008) Guidance for Industry and FDA Staff - Information for Manufacturers Seeking Marketing Clearance of Diagnostic Ultrasound Systems and Transducers. [Online]. Available: https://www.fda.gov/downloads/UCM070911.pdf

[49] ——, "Radio Frequency Wireless Technology in Medical Devices - Guidance for Industry and FDA Staff," *Guidance Document*, 2013.

[50] FDA, *Firmware Update to Address Cybersecurity Vulnerabilities Identified in Abbott's (formerly St. Jude Medical's) Implantable Cardiac Pacemakers: FDA Safety Communication*, 2019.

[51] Femto. (2020) 100/200 MHZ Wideband Voltage Amplifier Series DHPVA. [Online]. Available: https://www.femto.de/en/products/voltage-amplifiers/variable-gain-100-200-mhz-dhpva.html

[52] C. Fu, X. Du, L. Wu, Q. Zeng, A. Mohamed, and M. Guizani, "POKs Based Secure and Energy-Efficient Access Control for Implantable Medical Devices," in *Security and Privacy in Communication Networks*, 2019, pp. 105–125.

[53] K. Fu, "Medical device security," in *2016 USENIX Enigma Conference*. San Francisco, CA: USENIX Association, Jan. 2016.

[54] ——, "On the Technical Debt of Medical Device Security," in *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2015 Symposium*. National Academies Press, 2016.

[55] E. Gelenbe and Y. M. Kadioglu, "Battery attacks on sensors," in *International Symposium on Computer and Information Sciences, Security Workshop. Springer International Publishing*, 2018.

[56] I. Giechaskiel and K. Rasmussen, "Taxonomy and challenges of out-of-band signal injection attacks and defenses," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 645–670, 2019.

[57] I. Giechaskiel, Y. Zhang, and K. B. Rasmussen, "A framework for evaluating security in the presence of signal injection attacks," in *European Symposium on Research in Computer Security*. Springer, 2019, pp. 512–532.

[58] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: non-invasive security for implantable medical devices," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 2–13.

[59] Grand View Research, *Microelectronic Medical Implants Market Analysis Report by Product, by Technology, And Segment Forecasts, 2018 - 2025*, 2018. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/microelectronic-medical-implants-market

[60] A. Grisey, M. Heidmann, V. Letort, P. Lafitte, and S. Yon, "Influence of skin and subcutaneous tissue on high-intensity focused ultrasound beam: experimental quantification and numerical modeling," *Ultrasound in medicine & biology*, vol. 42, no. 10, pp. 2457–2465, 2016.

[61] T. Halevi and N. Saxena, "On pairing constrained wireless devices based on secrecy of auxiliary channels: the case of acoustic eavesdropping," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 97–108.

[62] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 129–142.

[63] P. Hasgall, F. Di Gennaro, C. Baumgartner, E. Neufeld, B. Lloyd, M. Gosselin, D. Payne, A. Klingenböck, and N. Kuster, "IT'IS Database for thermal and electromagnetic parameters of biological tissues. Version 4.0, May 15, 2018. DOI: 10.13099/VIP21000-04-0," IT'IS Foundation, Tech. Rep., 2018.

[64] X. Hei, X. Du, J. Wu, and F. Hu, "Defending resource depletion attacks on implantable medical devices," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. IEEE, 2010, pp. 1–5.

[65] High-Tech Bridge. (2015) Improper Access Control [CWE-284]. [Online]. Available: https://www.htbridge.com/vulnerability/improper-access-control.html

[66] J. S. Ho, A. J. Yeh, E. Neofytou, S. Kim, Y. Tanabe, B. Patlolla, R. E. Beygui, and A. S. Poon, "Wireless power transfer to deep-tissue microimplants," *Proceedings of the National Academy of Sciences*, vol. 111, no. 22, pp. 7974–7979, 2014.

[67] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen, "Opfka: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2274–2282.

[68] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security–a survey," *IEEE Internet of Things Journal*, 2017.

[69] IEC, "Functional safety of electrical/electronic/programmable electronic safety-related systems," *IEC 61508*, 2008.

[70] IEEE, "IEEE Standard for Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3 kHz to 300 GHz," *IEEE Std C95.1-2005 (Revision of IEEE Std C95.1-1991)*, pp. 1–238, 4 2006.

[71] ISO, "Information Technology - Security techniques - Entity authentication - Part 2: Mechanisms using symmetric encipherment algorithms," *ISO/IEC 9798-2:2008*, 2008.

[72] B. Jaafar, A. Soltan, J. Neasham, and P. Degenaar, "Wireless ultrasonic communication for biomedical injectable implantable device," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 4024–4027.

[73] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in *Proceedings of the 1999 Networks and distributed system security symposium*, 1999.

[74] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *ACM CCS*, 1999, pp. 28–36.

[75] K. Kadirvel, J. Carpenter, and B. Lum-Shue-Chan, "Power-management functions for energy harvesting," *EE Times*, 2012.

[76] Z. Kashani and M. Kiani, "Optimal ultrasonic pulse transmission for miniaturized biomedical implants," in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2019, pp. 1–4.

[77] Y. Kim, W. S. Lee, V. Raghunathan, N. K. Jha, and A. Raghunathan, "Vibration-based secure side channel for medical devices," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 32.

[78] L. Kuhlmann, K. Lehnertz, M. P. Richardson, B. Schelter, and H. P. Zaveri, "Seizure prediction—ready for a new era," *Nature Reviews Neurology*, vol. 14, no. 10, pp. 618–630, 2018.

[79] D. F. Kune, J. Backes, S. S. Clark, D. Kramer, M. Reynolds, K. Fu, Y. Kim, and W. Xu, "Ghost talk: Mitigating EMI signal injection attacks against analog sensors," in *2013 IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 145–159.

[80] P. Lafourcade and M. Puys, "Performance evaluations of cryptographic protocols verification tools dealing with algebraic properties," in *International*

Symposium on Foundations and Practice of Security. Springer, 2015, pp. 137–155.

[81] A. Lafrance. (2014) Who Killed the Rechargeable Pacemaker? [Online]. Available: https://www.theatlantic.com/health/archive/2014/02/who-killed-the-rechargeable-pacemaker/283365/

[82] M. E. Leckrone and V. T. Cutolo Jr, "Multi-mode microprocessor-based programmable cardiac pacer," 12 1984, US Patent 4,485,818.

[83] J.-L. Letouzey, "The SQALE method for evaluating technical debt," in *2012 Third International Workshop on Managing Technical Debt (MTD)*. IEEE, 2012, pp. 31–36.

[84] C. Li, A. Raghunathan, and N. K. Jha, "Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system," in *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*. IEEE, 2011, pp. 150–156.

[85] P. Li and R. Bashirullah, "A Wireless Power Interface for Rechargeable Battery Operated Medical Implants," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 10, pp. 912–916, 2007.

[86] P. Lindqvist *et al.*, "Compression and storage of medical data in pacemakers," *Master's Thesis, Royal Institute of Technology Stockholm, Sweden*, 2005.

[87] J.-W. Liu, M. Al Ameen, and K.-S. Kwak, "Secure wake-up scheme for WBANs," *IEICE transactions on communications*, vol. 93, no. 4, pp. 854–857, 2010.

[88] P. Luckett, J. McDonald, and W. Glisson, "Attack-Graph Threat Modeling Assessment of Ambulatory Medical Devices," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[89] A. L. Mansano, Y. Li, S. Bagga, and W. A. Serdijn, "An Autonomous Wireless Sensor Node With Asynchronous ECG Monitoring in 0.18 μ m CMOS," *IEEE transactions on biomedical circuits and systems*, vol. 10, no. 3, pp. 602–611, 2016.

[90] D. Mao, L. Zhang, X. Li, and D. Mu, "Trusted authority assisted three-factor authentication and key agreement protocol for the implantable medical system," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[91] Maria Guerra. (2016) Can Supercapacitors Surpass Batteries for Energy Storage? [Online]. Available: https://www.electronicdesign.com/power/can-supercapacitors-surpass-batteries-energy-storage

[92] R. Mariani. (2011) Deliverable D2.1: Application Analysis Guide. [Online]. Available: http://www.desyre.eu/?q=%3Cdeliverables%3E

[93] E. Marin, E. Argones-Rúa, D. Singelée, and B. Preneel, "A survey on physiological-signal-based security for medical devices." *IACR Cryptology ePrint Archive*, vol. 2016, p. 867, 2016.

[94] E. Marin, D. Singelée, F. D. Garcia, T. Chothia, R. Willems, and B. Preneel, "On the (in) security of the latest generation implantable cardiac defibrillators and how to secure them," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*. ACM, 2016, pp. 226–236.

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

[96] E. Marín Fàbregas, "Security and Privacy of Implantable Medical Devices," Ph.D. dissertation, KU Leuven, Belgium, 2018.

[97] E. Martin, J. Jaros, and B. E. Treeby, "Experimental validation of k-wave: Nonlinear wave propagation in layered, absorbing fluid media," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 67, no. 1, pp. 81–91, 2019.

[98] G. C. Martins, A. Urso, A. Mansano, Y. Liu, and W. A. Serdijn, "Energy-efficient low-power circuits for wireless energy and data transfer in iot sensor nodes," *CoRR*, vol. abs/1704.08910, 2017. [Online]. Available: http://arxiv.org/abs/1704.08910

[99] R. Mayrhofer and H. Gellersen, "On the security of ultrasound as out-of-band channel," in *2007 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2007, pp. 1–6.

[100] Medtronic, *GEM® III VR 7231 Implantable Cardioverter Defibrillator - System Reference Guide*, 2001.

[101] ——, *Kappa® 400 series and DX2 pacemakers Model 9952 - Volume II, Pacemaker Reference Guide*, 2001.

[102] Medtronic, *Activa PC - Implant manual*, 2008.

[103] Medtronic, *Azure™ S SR MRI SureScan™ W3SR01 - Device Manual*, 2017.

[104] Medtronic. (2018) RestoreUltra SureScan MRI Neurostimulator - Spinal Cord Stimulation. [Online]. Available: http://www.medtronic.com/us-en/healthcare-professionals/products/neurological/spinal-cord-stimulation-systems/restoreultra-surescan-mri-neurostimulator.html

[105] Medtronic, *SECURITY BULLETIN – Conexus™ Telemetry and Monitoring Accessories*, 2019.

[106] N. Mehta, "When to Consider Getting a Rechargeable SCS," *Veritas Health*, 2018.

[107] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The TAMARIN prover for the symbolic analysis of security protocols," in *International Conference on Computer Aided Verification*. Springer, 2013, pp. 696–701.

[108] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 2018.

[109] Microsemi, *ZL70103 Medical Implantable RF Transceiver - Datasheet, Revision 2*, 2015.

[110] D. Mishra, S. De, and K. R. Chowdhury, "Charging time characterization for wireless rf energy transfer," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 4, pp. 362–366, 2015.

[111] H. G. Mond, "The development of pacemaker programming: Memories from a bygone era," *Heart, Lung and Circulation*, 2020.

[112] NeuroPace, *RNS® System User Manual*, 2019.

[113] Olympus, *Panametrics® Ultrasonic Transducers - Wedges, Cables, Test Blocks*, 2016.

[114] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[115] F. Palomba, D. Di Nucci, A. Panichella, A. Zaidman, and A. De Lucia, "On the impact of code smells on the energy consumption of mobile applications," *Information and Software Technology*, vol. 105, pp. 43–55, 2019.

[116] C.-S. Park, "Security mechanism based on hospital authentication server for secure application of implantable medical devices," *BioMed research international*, vol. 2014, 2014.

[117] C. Patrick and P. Schaumont, "The role of energy in the lightweight cryptographic profile," in *NIST Lightweight Cryptography Workshop*, 2016.

[118] S. Peter, B. Pratap Reddy, F. Momtaz, and T. Givargis, "Design of secure ecg-based biometric authentication in body area sensor networks," *Sensors*, vol. 16, no. 4, p. 570, 2016.

[119] Pike and I. Fischer, *Communications Regulation*, ser. Communications Regulation. Pike & Fischer, 2003.

[120] G. Pinto, F. Soares-Neto, and F. Castor, "Refactoring for energy efficiency: A reflection on the state of the art," in *2015 IEEE/ACM 4th International Workshop on Green and Sustainable Software*. IEEE, 2015, pp. 29–35.

[121] V. Pournaghshband, M. Sarrafzadeh, and P. Reiher, "Securing legacy mobile medical devices," in *International Conference on Wireless Mobile Communication and Healthcare*. Springer, 2012, pp. 163–172.

[122] D. Prutchi, "St. Jude's (ANS) Rechargeable Spinal Cord Stimulators Eon and Eon Mini," 2012.

[123] F. Putz, F. Álvarez, and J. Classen, "Acoustic integrity codes: secure device pairing using short-range acoustic communication," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020, pp. 31–41.

[124] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 410–419.

[125] H. Rathore, A. Mohamed, A. Al-Ali, X. Du, and M. Guizani, "A review of security challenges, attacks and resolutions for wireless medical devices," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1495–1501.

[126] H. Rathore, C. Fu, A. Mohamed, A. Al-Ali, X. Du, M. Guizani, and Z. Yu, "Multi-layer security scheme for implantable medical devices," *Neural Computing and Applications*, pp. 1–14, 2018.

[127] K. Rindell, K. Bernsmed, and M. G. Jaatun, "Managing security in software: Or: How i learned to stop worrying and manage the security technical debt," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1–8.

[128] M. Roe, "Cryptography and evidence," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-780, May 2010. [Online]. Available: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-780.pdf

[129] M. Roeschlin, I. Martinovic, and K. B. Rasmussen, "Device pairing at the touch of an electrode," in *Proceedings of NDSS*, vol. 18, 2018.

[130] M. Rostami, W. Burleson, F. Koushanfar, and A. Juels, "Balancing security and utility in medical devices?" in *Proceedings of the 50th Annual Design Automation Conference*. ACM, 2013, p. 13.

[131] M. Rostami, A. Juels, and F. Koushanfar, "Heart-to-heart (H2H): authentication for implanted medical devices," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 1099–1112.

[132] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "Sok: Security and privacy in implantable medical devices and body area networks," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 524–539.

[133] A. P. Sample, D. J. Yeager, P. S. Powledge, and J. R. Smith, "Design of a passively-powered, programmable sensing platform for uhf rfid systems," in *RFID, 2007. IEEE International Conference on*. IEEE, 2007, pp. 149–156.

[134] G. E. Santagati, N. Dave, and T. Melodia, "Design and performance evaluation of an implantable ultrasonic networking platform for the internet of medical things," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 29–42, 2020.

[135] G. E. Santagati and T. Melodia, "An implantable low-power ultrasonic platform for the internet of medical things," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[136] P. Schaumont, B. Yuce, K. Pabbuleti, and D. Mane, "Secure authentication with energy-harvesting: A multi-dimensional balancing act," *Sustainable Computing: Informatics and Systems*, vol. 12, pp. 83–95, 2016.

[137] B. Schneier, "Attack trees," *Dr. Dobb's journal*, vol. 24, no. 12, pp. 21–29, 1999.

[138] R. M. Seepers, J. H. Weber, Z. Erkin, I. Sourdis, and C. Strydis, "Secure key-exchange protocol for implants using heartbeats," in *Proceedings of the ACM International Conference on Computing Frontiers*. ACM, 2016, pp. 119–126.

[139] R. M. Seepers, "Implantable Medical Devices: Device security and emergency access," Ph.D. dissertation, Erasmus University Medical Center, Rotterdam, Netherlands, 12 2016.

[140] D. Seo, R. M. Neely, K. Shen, U. Singhal, E. Alon, J. M. Rabaey, J. M. Carmena, and M. M. Maharbiz, "Wireless recording in the peripheral nervous system with ultrasonic neural dust," *Neuron*, vol. 91, no. 3, pp. 529–539, 2016.

[141] N. Shevchenko, T. A. Chick, P. O'Riordan, T. P. Scanlon, and C. Woody, "Threat Modeling: A Summary of Available Methods," Software Engineering Institute - Carnegie Mellon University, Tech. Rep., 2018.

[142] M. Siavvas, D. Tsoukalas, M. Janković, D. Kehagias, A. Chatzigeorgiou, D. Tzovaras, N. Aničić, and E. Gelenbe, "An empirical evaluation of the relationship between technical debt and software security," in *9th International Conference on Information Society and Technology*, 2019.

[143] M. G. Siavvas, K. C. Chatzidimitriou, and A. L. Symeonidis, "Qatch-an adaptive framework for software product quality assessment," *Expert Systems with Applications*, vol. 86, pp. 350–366, 2017.

[144] M. A. Siddiqi, C. Doerr, and C. Strydis, "IMDfence: Architecting a Secure Protocol for Implantable Medical Devices," *IEEE Access*, vol. 8, pp. 147 948–147 964, 2020.

[145] M. A. Siddiqi, R. H. S. H. Beurskens, P. Kruizinga, C. I. De Zeeuw, and C. Strydis, "Securing Implantable Medical Devices Using Ultrasound Waves," *IEEE Access*, vol. 9, pp. 80 170–80 182, 2021.

[146] M. A. Siddiqi, R. M. Seepers, M. Hamad, V. Prevelakis, and C. Strydis, "Attack-tree-based Threat Modeling of Medical Implants," in *PROOFS 2018. 7th International Workshop on Security Proofs for Embedded Systems*, ser. Kalpa Publications in Computing, vol. 7. EasyChair, 2018, pp. 32–49.

[147] M. A. Siddiqi, W. A. Serdijn, and C. Strydis, "Zero-Power Defense Done Right: Shielding IMDs from Battery-Depletion Attacks," *Journal of Signal Processing Systems*, pp. 1–17, 2020.

[148] M. A. Siddiqi and C. Strydis, "IMD Security vs. Energy: Are we tilting at windmills?: POSTER," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*. ACM, 2019, pp. 283–285.

[149] ——, "Towards Realistic Battery-DoS Protection of Implantable Medical Devices," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*. ACM, 2019, pp. 42–49.

[150] M. A. Siddiqi, C. Strydis, and C. I. De Zeeuw, "Implantable Medical Device and Control Device Therefor," Jun. 29 2021, NL Patent App. N2028563 & N2028564.

[151] M. A. Siddiqi, A.-A. Tsintzira, G. Digkas, M. Siavvas, and C. Strydis, "Adding Security to Implantable Medical Devices: Can We Afford It?" in *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '21. USA: Junction Publishing, 2021, p. 67–78.

[152] Silicon Labs. (2013) Using EFM32 in Energy Harvesting Applications AN0061 - Application Note. [Online]. Available: https://www.silabs.com/documents/public/application-notes/AN0061.pdf

[153] ——, *EFM32 Tiny Gecko 11 Family - Reference Manual*, 2018.

[154] ——, *UG103.6: Bootloader Fundamentals*, 2020.

[155] F. Simon, J. P. Martinez, P. Laguna, B. van Grinsven, C. Rutten, and R. Houben, "Impact of sampling rate reduction on automatic ecg delineation," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 2587–2590.

[156] J. Sorber, M. Shin, R. Peterson, C. Cornelius, S. Mare, A. Prasad, Z. Marois, E. Smithayer, and D. Kotz, "An amulet for trustworthy wearable mhealth," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*. ACM, 2012, p. 7.

[157] St. Jude Medical, *Product Performance Report - Cardiac Rhythm Management*, 2006.

[158] ——, *Clinician Programmer App For Spinal Cord Stimulation Systems Model 3874 - Clinician's Manual*, 2015.

[159] ——, *FAQs - Merlin.net^{TM} Patient Care Network (PCN) 8.0 Q&A*, 2015.

[160] ——, *Confirm Rx^{TM} Model DM3500 Insertable Cardiac Monitor - User's Guide*, 2016.

[161] ——, *Ellipse, Fortify Assura ICD, Quadra Assura, Quadra Assura MP, Unify Assura CRT-D User Manual*, 2017.

[162] ——, *Proclaim^{TM} Implantable Pulse Generator - Clinician's Manual*, 2017.

[163] C. Strydis, "Universal Processor Architecture for Biomedical Implants: The SiMS Project," Ph.D. dissertation, Delft University of Technology, Delft, Netherlands, 3 2011.

[164] C. Strydis, R. M. Seepers, P. Peris-Lopez, D. Siskos, and I. Sourdis, "A system architecture, processor, and communication protocol for secure implants," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 10, no. 4, p. 57, 2013.

[165] R. V. Taalla, M. S. Arefin, A. Kaynak, and A. Z. Kouzani, "A review on miniaturized ultrasonic wireless power transfer to implantable medical devices," *IEEE access*, vol. 7, pp. 2092–2106, 2018.

[166] W. O. Tatum, "Ellen R. Grass Lecture: Extraordinary EEG," *The Neurodiagnostic Journal*, vol. 54, no. 1, pp. 3–21, 2014.

[167] C. R. Taylor, K. Venkatasubramanian, and C. A. Shue, "Understanding the security of interoperable medical devices using attack graphs," in *Proceedings of the 3rd international conference on High confidence networked systems*. ACM, 2014, pp. 31–40.

[168] Texas Instruments, *MSP430F15x, MSP430F16x, MSP430F161x Mixed Signal Microcontroller - Datasheet*, 2002.

[169] ——, *MSP430FG47x Mixed Signal Microcontroller - Datasheet*, 2008.

[170] The AVISPA Team, *AVISPA v1.1 User Manual*, 2006.

[171] N. O. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun, "On limitations of friendly jamming for confidentiality," in *2013 IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 160–173.

[172] Y. Todo, "Integral cryptanalysis on full MISTY1," *Journal of Cryptology*, vol. 30, no. 3, pp. 920–959, 2017.

[173] W. J. Tomlinson, S. Banou, C. Yu, M. Stojanovic, and K. R. Chowdhury, "Comprehensive survey of galvanic coupling and alternative intra-body communication technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1145–1164, 2018.

[174] B. E. Treeby and B. T. Cox, "k-wave: Matlab toolbox for the simulation and reconstruction of photoacoustic wave fields," *Journal of biomedical optics*, vol. 15, no. 2, p. 021314, 2010.

[175] U.S. Bureau of Labor Statistics. (2019) May 2019 National Occupational Employment and Wage Estimates United States. [Online]. Available: https://www.bls.gov/oes/current/oes_nat.htm

[176] J. van den Breekel, "A security evaluation and proof-of-concept relay attack on dutch emv contactless transactions," *Master's thesis*, 2014.

[177] J. van den Breekel, D. A. Ortiz-Yepes, E. Poll, and J. de Ruiter, "Emv in a nutshell," KPMG, Tech. Rep., 2016.

[178] M. N. van Dongen, A. Karapatis, L. Kros, O. E. Rooda, R. M. Seepers, C. Strydis, C. I. De Zeeuw, F. E. Hoebeek, and W. A. Serdijn, "An implementation of a wavelet-based seizure detection filter suitable for realtime closed-loop epileptic seizure suppression," in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*. IEEE, 2014, pp. 504–507.

[179] W. Verkruysse, L. O. Svaasand, and J. S. Nelson, "Remote plethysmographic imaging using ambient light." *Optics express*, vol. 16, no. 26, pp. 21 434–21 445, 2008.

[180] A. Vladišauskas and L. Jakevičius, "Absorption of ultrasonic waves in air," *Ultragarsas*, vol. 50, no. 1, pp. 46–49, 2004.

[181] M. Wazid, A. K. Das, N. Kumar, M. Conti, and A. V. Vasilakos, "A novel authentication and key agreement scheme for implantable medical devices deployment," *IEEE journal of biomedical and health informatics*, vol. 22, no. 4, pp. 1299–1309, 2018.

[182] M. S. Wegmüller, "Intra-body communication for biomedical sensor networks," Ph.D. dissertation, ETH Zurich, 2007.

[183] L. Wu, X. Du, M. Guizani, and A. Mohamed, "Access control schemes for implantable medical devices: A survey," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1272–1283, 2017.

[184] J. Xu, K. K. Venkatasubramanian, and V. Sfyrla, "A methodology for systematic attack trees generation for interoperable medical devices," in *2016 Annual IEEE Systems Conference (SysCon)*. IEEE, 2016, pp. 1–7.

[185] K.-W. Yang, K. Oh, and S. Ha, "Challenges in scaling down of free-floating implantable neural interfaces to millimeter scale," *IEEE Access*, vol. 8, pp. 133 295–133 320, 2020.

[186] Q. Yang, S. Mai, Y. Zhao, Z. Wang, C. Zhang, and Z. Wang, "An on-chip security guard based on zero-power authentication for implantable medical devices," in *Circuits and Systems (MWSCAS), 2014 IEEE 57th International Midwest Symposium on*. IEEE, 2014, pp. 531–534.

[187] G. Zheng, R. Shankaran, M. A. Orgun, L. Qiao, and K. Saleem, "Ideas and challenges for securing wireless implantable medical devices: A review," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 562–576, 2016.

[188] G. Zheng, W. Yang, C. Valli, L. Qiao, R. Shankaran, M. A. Orgun, and S. C. Mukhopadhyay, "Finger-to-heart (F2H): Authentication for wireless implantable medical devices," *IEEE journal of biomedical and health informatics*, 2018.

Acknowledgments

بسم الله الرحمن الرحيم الحمد لله و الصلوة و السلام على رسول الله
و على أزواجه و آله و أصحابه أجمعين إلى يوم الدين

First of all, I thank Allah, the Almighty, for everything.

There is no doubt in my mind that this work would never have reached its current state without the involvement and support of many people. Words cannot do justice in properly acknowledging them, but I will make an attempt nevertheless.

I will start by thanking Christos, who gave me the opportunity to pursue this research. Christos, I am grateful to you for allowing me sufficient space to explore independently, and consequently, grow intellectually. Your pedantic attention to details (or dare I say, OCD) in maintaining high quality has led to excellent publications. Moreover, thanks for your patience and bearing with me. You were always available to listen despite your swamped schedule. All in all, it was a genuine pleasure to work under your mentorship.

Chris, I am grateful to you for allowing me to pursue a multidisciplinary research at your vibrant department. Thanks for being so supportive even though there was very little overlap between my and the department's research trajectories.

Wouter, though you were not officially my supervisor, it felt like you were more than that. A significant portion of my PhD sprung out from our various meetings. Despite your busy schedule, you always found time to respond to my technical queries. I am not exaggerating when I say that you are a true definition of an *Educator*. Also, thanks for providing me the opportunity to deliver lectures in your courses at TU Delft, which has truly been an amazing educational experience.

I am also highly indebted to all my school and university teachers, especially those from Cadet College Kohat, Military College of Signals, NTNU Trondheim and the University of Southampton, for making me a more educated and better person. Furthermore, I cannot thank enough the folks at Silicon Labs Norway and Lahore University of Management Sciences for being a part of a truly enriching professional experience. Thanks to all my ex-managers and mentors, namely Shahid Masud, Øivind, Paul, Øyvind, Eirik, Arjan and Anubhav, for playing a significant role in my professional development, and for providing me with an invaluable skill set to pursue a doctoral degree.

I would like to convey my special thanks to my co-authors, namely Christian, Robert Seepers, Vassilis, Mohammad, Robert Beurskens, Pieter, Angeliki, Miltos and George, for contributing to my research and making it *a lot* better than what it would have been. Also, I am grateful to my colleagues: George, Rene, Jan-Harm, Angelo,

Sadaf, Farnaz, Harry, Mario, Bas K., Bas G., Bas v.H., Stephanie, Nikki, Nina and many others, for all their help and making the lab a vibrant place.

Now, I will talk about my biggest support system: my family. Dear Mama and Baba, I am here today because of your *tarbiyah* and prayers. I cannot thank you enough for your love and patience during my upbringing. *My Lord! Be merciful to them as they raised me when I was young (Al-Qur'an 17:24)*, Ameen. My dear wife Farwa, you were always there by my side during the highs and lows of PhD life. You were the one who pushed me over the line to start my PhD (which was my dream), and made sure that I finished it. *Our Lord! Grant us that our spouses and our offspring be a joy to our eyes, and do make us the leaders of the God-fearing (Al-Qur'an 25:74)*, Ameen. Omer, my brother, thank you for all the pep talks, support, pearls of wisdom and brotherly fights. I would also like to thank Nano, uncles (Taya, Mamoos and Phupha), aunts (Phuphos, Tai and Mumanis), cousins, sisters-in-law (Yusra and Sarah), Umar and rest of the close relatives, for their continuous support and prayers.

Besides family, I also have been blessed with wonderful friends, who are like siblings to me: Umair, Muqeet, Talha, Shahrukh, Wajid, Hammad, Saad, Qasim, and others, especially those from the *Kohatians* and *Signalians* clans. Thank you all for your unwavering support and counsel, and for bearing with me all these years. I am also grateful for the Iqbal family for helping me quickly settle in Rotterdam and for keeping us company.

Lastly, this PhD will be a constant reminder of my dearest ones who passed away; they were there at the start of my research – always rooting and praying for me – but could not witness its completion: my youngest brother Osman, parents-in-law Zarina and Afif, and uncles Sabir and Tahir. May Allah have mercy on them all, Ameen. In contrast, this PhD will also remind me of the two precious gifts from Allah: Hasan and Husain, for which I am eternally grateful.

Curriculum Vitae

Personal information

Name	Muhammad Ali Siddiqi
Date of birth	August 3rd 1987

Education

2017-2021 **Doctoral Program,** *Neuroscience department, Erasmus Medical Center, The Netherlands*
Promoter: Prof. Dr. Chris I. de Zeeuw

2010-2012 **MSc Embedded Computing Systems,** *Norwegian University of Science and Technology & University of Southampton, UK*
- MSc Project: *Analysing the effects of Standard Cell Utilization on the Post-layout timing for a high performance ARM CPU core using a 40 nm CMOS process*
- Average Grade: *A (Highest)*

2005-2009 **BE Electrical Engineering,** *National University of Sciences and Technology, Pakistan*
- BE Project: *Implementation of a Cognitive Radio System*
- CGPA: *3.95/4.0 (Highest)*

Additional courses

2017 **ASIP Designer Tutorial and Training,** *Synopsys ASIP University Day, Leuven, Belgium*

Awards

2012 MSc degree graduated with the **highest grade (A)**

2010-2012 Erasmus Mundus Master **Scholarship**

2009 **President's Gold Medal** for obtaining the highest CGPA in the BE studies

2005-2009 National University of Sciences and Technology (NUST) **Merit Scholarship**

Other academic experiences

- **Supervised 4 MSc theses** (students from TU Delft, the Netherlands)
- **Supervised 2 BSc projects** (10 students from TU Delft, the Netherlands)
- **Gave lectures on the security & reliability of IMDs** (TU Delft, 2017-2021, courses: *EE4555 Active Implantable Biomedical Microsystems & TM12003 Electrostimulation of Neurophysiological systems*)
- Presented work at **3 international conferences**
- **Peer-reviewed 12 scientific articles** for various renowned journals and conferences

Working experience

2012-2017 **Design Engineer,** *Silicon Labs Norway*
• Specification, design and verification of digital peripherals in microcontroller and radio system-on-chip products. Examples include the Protocol Timer and Cryotimer, which are part of the EFR32^{TM} Wireless Gecko SoCs.
• Tasks: Top-level (system) verification, low-power digital design and verification, RTL synthesis, static timing analysis, technical documentation

2009-2010 **Research Assistant,** *Department of Computer Science, Lahore University of Management Sciences (LUMS), Pakistan*
• Implementation of efficient sample-rate converters for software defined radios
• Implementation of adaptive delta-sigma modulators

List of publications included in this thesis

- **M. A. Siddiqi**, R. H. S. H. Beurskens, P. Kruizinga, C. I. De Zeeuw, and C. Strydis, "Securing Implantable Medical Devices Using Ultrasound Waves," *IEEE Access*, vol. 9, pp. 80 170–80 182, 2021.

- **M. A. Siddiqi**, C. Strydis, and C. I. De Zeeuw, "Implantable Medical Device and Control Device Therefor," Jun. 29 2021, *NL Patent App.* N2028563 & N2028564.

- **M. A. Siddiqi**, A.-A. Tsintzira, G. Digkas, M. Siavvas, and C. Strydis, "Adding Security to Implantable Medical Devices: Can We Afford It?" in *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '21. USA: Junction Publishing, 2021, p. 67–78.

- **M. A. Siddiqi**, C. Doerr, and C. Strydis, "IMDfence: Architecting a Secure Protocol for Implantable Medical Devices," *IEEE Access*, vol. 8, pp. 147 948–147 964, 2020.

- **M. A. Siddiqi**, W. A. Serdijn, and C. Strydis, "Zero-Power Defense Done Right: Shielding IMDs from Battery-Depletion Attacks," *Journal of Signal Processing Systems*, pp. 1–17, 2020.

- **M. A. Siddiqi** and C. Strydis, "Towards Realistic Battery-DoS Protection of Implantable Medical Devices," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*. ACM, 2019, pp. 42–49.

- **M. A. Siddiqi** and C. Strydis, "IMD Security vs. Energy: Are we tilting at windmills?: POSTER," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*. ACM, 2019, pp. 283–285.

- **M. A. Siddiqi**, R. M. Seepers, M. Hamad, V. Prevelakis, and C. Strydis, "Attack-tree-based Threat Modeling of Medical Implants," in *PROOFS 2018. 7th International Workshop on Security Proofs for Embedded Systems*, ser. Kalpa Publications in Computing, vol. 7. EasyChair, 2018, pp. 32–49.

Summary

Implantable Medical Devices (IMDs), such as pacemakers, cardioverter defibrillators, neurostimulators etc., belong to a class of highly life-critical, resource-constrained, deeply embedded systems out there. Their gradual conversion to wirelessly accessible devices in recent years has made them amenable to security attacks from malicious entities. This can lead to serious issues for the implant host, such as private-data theft and physical harm and is also reflected in numerous, successful ethical-hacking attempts, all made possible due to the absence of proper security provisions in these devices. This situation is a direct result of the lack of security standardization in this domain, the highly resource-constrained nature of IMDs, and the steep (re)certification costs of adding security, among others. In this thesis, I first provide an overview of the IMD threat landscape by employing attack trees as a threat-analysis tool, and subsequently highlight the areas of utmost importance in terms of security. Next, I provide a systematization of protection mechanisms against battery-depletion attacks, which are among the easiest attacks to mount from the attacker's perspective. Then, I propose a security protocol for IMD ecosystems that satisfies a comprehensive portfolio of security requirements, which includes availability, non-repudiation, access control, entity authentication, remote monitoring and system scalability. I subsequently describe a novel device-pairing scheme that is based on ultrasound communication. This scheme establishes trust between the IMD and an external device as well as protects against battery-depletion attacks. In the last part of this thesis, I take a different tack and assess the economic viability of adding security to IMD systems using the concept of Technical Debt. The analysis yields worrying signs down the road for implant manufacturers, as costs do not appear to scale. Clearly, a new approach to secure IMD design is called for.

Samenvatting

Implanteerbare medische apparaten (IMD's), zoals pacemakers, cardioverter-defibrillators, neurostimulatoren, enz., behoren tot de meest levenskritische, resourcebeperkte, diep geïntegreerde systemen die er bestaan. De geleidelijke verschuiving naar draadloze toegang voor deze apparaten in de afgelopen jaren heeft ze vatbaar gemaakt voor beveiligingsaanvallen van kwaadwillenden. Dit kan leiden tot ernstige problemen voor de drager van het implantaat, zoals diefstal van privégegevens en schade aan de gezondheid. De talrijke succesvolle pogingen tot ethische hacking, allemaal mogelijk gemaakt door het ontbreken van goede beveiligingsvoorzieningen in deze apparaten, hebben dit laten zien. Deze situatie is onder andere een direct gevolg van het gebrek aan standaardisatie van beveiliging in dit domein, de zeer beperkte rekenkracht van IMD's en de hoge (her)certificeringskosten van het toevoegen van beveiliging. In dit proefschrift geef ik eerst een overzicht van het IMD-dreigingslandschap, door attack trees te gebruiken als een manier om dreigingen te analyseren. Vervolgens belicht ik de gebieden die het belangrijkst zijn als het gaat om beveiliging. Vervolgens geef ik een systematisch overzicht van mechanismen die beschermen tegen aanvallen gericht op het leegraken van de batterij. Deze aanvallen behoren, vanuit het perspectief van de aanvaller, tot de gemakkelijkste aanvallen. Vervolgens stel ik een beveiligingsprotocol voor IMD-ecosystemen voor dat voldoet aan een veelomvattend scala aan beveiligingsvereisten, waaronder beschikbaarheid, onweerlegbaarheid, toegangscontrole, entiteitsverificatie, toezicht op afstand en systeemschaalbaarheid. Vervolgens beschrijf ik een nieuw apparaatpairingsschema dat is gebaseerd op ultrasone communicatie. Dit schema zorgt voor vertrouwen tussen de IMD en een extern apparaat en beschermt tegen aanvallen gericht op het leegraken van de batterij. In het laatste deel van dit proefschrift sla ik een andere weg in en beoordeel de economische haalbaarheid van het toevoegen van beveiliging aan IMD-systemen. Dit doe ik met behulp van het concept van technische schuld. De analyse laat zien dat er verontrustende signalen zijn voor implantaatfabrikanten, aangezien de kosten niet lijken op te schalen. Het is duidelijk dat er behoefte is aan een nieuwe benadering voor het ontwerpen van veilige IMD's.