Securing Wireless Neurostimulators

Eduard Marin[1], Dave Singelée[1], Bohan Yang[1], Vladimir Volski[2], Guy A. E. Vandenbosch[2]
Bart Nuttin[3] and Bart Preneel[1]
[1] imec-COSIC, KU Leuven, Belgium
{firstname.lastname}@esat.kuleuven.be
[2] ESAT-TELEMIC, KU Leuven, Belgium
{vladimir.volski,guy.vandenbosch}@kuleuven.be
[3] Neurosurgery, UZ Leuven, Belgium
{bart.nuttin}@kuleuven.be

ABSTRACT

Implantable medical devices (IMDs) typically rely on proprietary protocols to wirelessly communicate with external device programmers. In this paper, we fully reverse engineer the proprietary protocol between a device programmer and a widely used commercial neurostimulator from one of the leading IMD manufacturers. For the reverse engineering, we follow a black-box approach and use inexpensive hardware equipment. We document the message format and the protocol state-machine, and show that the transmissions sent over the air are neither encrypted nor authenticated. Furthermore, we conduct several software radio-based attacks that could compromise the safety and privacy of patients, and investigate the feasibility of performing these attacks in real scenarios.

Motivated by our findings, we propose a security architecture that allows for secure data exchange between the device programmer and the neurostimulator. It relies on using a patient's physiological signal for generating a symmetric key in the neurostimulator, and transporting this key from the neurostimulator to the device programmer through a secret out-of-band (OOB) channel. Our solution allows the device programmer and the neurostimulator to agree on a symmetric session key without these devices needing to share any prior secrets; offers an effective and practical balance between security and permissive access in emergencies; requires only minor hardware changes in the devices; adds minimal computation and communication overhead; and provides forward and backward security. Finally, we implement a proof-of-concept of our solution.

CCS CONCEPTS

• **Security and privacy → Embedded systems security**; *Key management*;

KEYWORDS

Proprietary wireless communication protocol, black-box reverse engineering, low-cost randomness source, security architecture.

ACM Reference format:
Eduard Marin[1], Dave Singelée[1], Bohan Yang[1], Vladimir Volski[2], Guy A. E. Vandenbosch[2] Bart Nuttin[3] and Bart Preneel[1] [1] imec-COSIC, KU Leuven, Belgium {firstname.lastname}@esat.kuleuven.be [2] ESAT-TELEMIC, KU Leuven, Belgium {vladimir.volski,guy.vandenbosch}@kuleuven.be [3] Neurosurgery, UZ Leuven, Belgium {bart.nuttin}@kuleuven.be . 2018. Securing Wireless Neurostimulators. In *Proceedings of Eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, March 19–21, 2018 (CODASPY '18),* 12 pages.
https://doi.org/10.1145/3176258.3176310

1 INTRODUCTION

In the US, chronic pain already affects more people than those suffering from diabetes, heart disease and cancer altogether [1]. In recent years, the number of people with movement disorders such as Parkinson's disease or essential tremor has also increased. It is estimated that seven to ten million people worldwide are living with Parkinson's disease [6]. Many of these problems can be relieved with neurostimulators that deliver controlled electrical signals to specific targeted areas in the patient's brain.

The newest generations of neurostimulators often include wireless capabilities that enable remote monitoring and reprogramming through an external device programmer. While the wireless interface enables more flexible and personalized treatments for patients, it also opens the door for adversaries to conduct software radio-based attacks. If strong security mechanisms are not in place, adversaries could send malicious commands to the neurostimulator in order to deliver undesired electrical signals to the patient's brain. For example, adversaries could change the settings of the neurostimulator to increase the voltage of the signals that are continuously delivered to the patient's brain. This could prevent the patient from speaking or moving, cause irreversible damage to his brain, or even worse, be life-threatening.

Beyond attacks that can endanger the patient's safety, there are other attacks that can, for example, compromise the patient's privacy. On the one hand, adversaries could leverage the wireless nature of the communication to intercept the data transmitted over the air. The transmitted data is personal data, and some of it is sensitive medical data. On the other hand, a more sophisticated form of privacy attack would be to use signals extracted from the

CODASPY '18, March 19–21, 2018, Tempe, AZ, USA

ACM ISBN 978-1-4503-5632-9/18/03…$15.00
https://doi.org/10.1145/3176258.3176310

brain to make inferences about patients. While this is currently not possible, future generations of neurostimulators will use information extracted from patients brain signals for the development of more precise and effective therapies. In that case, adversaries could capture and analyze brain signals such as the P-300 wave, a brain response that begins 300 ms after a stimulus is presented to a subject. The P-300 wave shows the brain's ability to recognize familiar information. Martinovic et al. performed a side-channel attack on a Brain Computer Interface (BCI) while connected to several subjects, and showed that the P-300 wave can leak sensitive personal information such as passwords, PINs, whether a person is known to the subject, or even reveal emotions and thoughts [29]. All the attacks described above clearly show the need for analyzing the security and privacy of neurostimulators.

1.1 Problem statement and challenges

Several papers have demonstrated that implantable medical devices (IMDs) with wireless capabilities often lack strong security mechanisms. Halperin et al. were the first to identify some of the potential security and privacy threats on IMDs [19]. Hei et al. proposed simple yet effective denial-of-service (DoS) attacks against IMDs that cannot be prevented with traditional cryptographic approaches. The goal of these attacks is to deplete the IMD's resources such that the battery lifetime is reduced from several years to a few weeks [21]. These attacks are similar to the sleep deprivation torture attack proposed by Stajano and Anderson [38]. In 2008, Halperin et al. analyzed the proprietary protocol between a device programmer and an implantable cardioverter defibrillator (ICD) over a short-range communication channel (less than 10 cm) [20]. Because of the lack of security mechanisms, they were able to realize various attacks by replaying past transmissions sent by legitimate device programmers. Marin et al. fully reverse engineered the proprietary protocol between a device programmer and a latest generation ICD over a long-range communication channel (from 2 to 5 m) [27]. They also showed that it is possible to conduct attacks using only inexpensive hardware equipment without needing to be close to the patient. Similarly, Li et al. were able to emulate legitimate remote controls to perform attacks against insulin pumps [14]. Marin et al. extended the previous work by reverse engineering the proprietary protocol between an insulin pump and all its potential peripherals [28]. In previous work, Denning et al. and Rushanan et al. highlighted the need for evaluating the security and privacy of neurostimulators [15, 35]. In this paper, we tackle this problem and investigate the security of the proprietary protocol between a device programmer and a widely used commercial neurostimulator.

Securing the communication between IMDs and device programmers is a non-trivial task. Firstly, IMDs are resource-constrained devices with tight power and computational constraints, and a limited battery capacity. Furthermore, IMDs lack input and output interfaces, such as a keyboard or a screen, and cannot be physically accessed once they are implanted. Secondly, IMDs need to satisfy several important requirements for proper functioning, such as reliability, availability and safety. Adding security mechanisms into IMDs is challenging due to inherent tensions between some of these requirements and the desirable security and privacy goals. For example, IMDs should provide permissive access control such that doctors can access the IMD in emergencies. A small delay when contacting care providers for device-specific cryptographic keys could prevent the patient from receiving care on time. However, if access control policies are not sufficiently strong, IMDs could be exposed to attacks. This shows the need for designing security solutions that achieve an effective balance between security and permissive access control in emergencies.

To bootstrap a secure communication channel between the IMD and the device programmer, cryptographic keys first need to be securely initialized and shared. This can be achieved using traditional approaches based on symmetric or public-key cryptography. For example, one could pre-install a device-specific symmetric key in each IMD during manufacturing, or use a key diversification protocol [20, 27]. However, as explained above, it may not be possible for medical personnel to contact care providers on the fly for requesting device-specific cryptographic keys. If a key diversification protocol is used, storing the master key in tamper-proof hardware in all device programmers would bring substantial security risks; if the master key is ever compromised, the security of millions of IMDs would be at risk. Instead, one could opt for storing the master key in the cloud but this is not a viable option since device programmers are required to operate at all times, including during Internet or cloud provider outages. Another solution would be to pre-install a public-private key pair in each IMD, and use any secure key transport or key agreement protocol. This approach would require a robust, worldwide infrastructure that keeps an updated list of trustworthy device programmers as well as a revocation mechanism that ensures that IMDs cannot communicate with untrustworthy device programmers [9]. Unfortunately, having such a robust worldwide infrastructure is difficult, and IMDs do not have an Internet connection to periodically get a certificate revocation list (CRL) or sufficient memory to store all the necessary certificates. This shows that traditional solutions may not be applicable for IMDs due to their unique functional requirements and limited resources.

1.2 Contributions

To the best of our knowledge, this is the first paper that evaluates the security of the wireless communication protocol between a device programmer and a widely used neurostimulator from one of the leading IMD manufacturers. In addition, we propose a practical and efficient security architecture that enables the device programmer and the neurostimulator to create a secure communication channel. In detail, our main contributions can be summarized as follows:

- **Security analysis**. We describe the process of how to reverse engineer the proprietary protocol between the device programmer and the neurostimulator. We follow a *black-box reverse engineering* approach and use inexpensive commercial hardware equipment.
- **Software radio-based attacks**. We assess the feasibility and demonstrate software radio-based attacks on neurostimulators. We also elaborate on how adversaries can overcome some of the limitations and challenges of these attacks.
- **Low-cost source of randomness for neurostimulators**. We explore the possibility of using a patient's brain physiological signal as a randomness source for generating keys,

and detail the process of extracting entropy from it. We evaluate our technique using real data extracted from the brain of 22 mice.

- **Security architecture for neurostimulators.** We present a complete security architecture that includes the generation of random session keys, the secure transportation of these keys from the neurostimulator to the device programmer and the cryptographic protocols to secure the communication flow.

Disclosure of results. We followed the principle of responsible disclosure and contacted the manufacturer six months before publishing our results. After discussing our findings with the manufacturer, we chose not to disclose the full details of the obtained results since this could help someone to mount these attacks on neurostimulators used by patients.

Paper outline. The remainder of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 describes the devices that comprise the neurostimulation system, whereas our laboratory setup is shown in Sect. 4. Section 5 details the process of reverse engineering the proprietary protocol between the device programmer and the neurostimulator to discover the message format and the protocol state-machine. Section 6 shows several software radio-based attacks that we are able to conduct on the neurostimulator, while a security architecture to preclude these attacks is proposed in Sect. 7. Section 8 discusses possible limitations of our solution and provides some directions for future work. Section 9 gives concluding remarks.

2 RELATED WORK

There are two main categories of countermeasures to secure the communication between the IMD and the device programmer: (i) those based on using an external device as a proxy, and (ii) those relying on an out-of-band channel to allow the devices securely agree on a cryptographic key.

2.1 External devices

Gollakota et al. proposed an external device known as "shield", that jams the messages to/from the IMD to prevent others from decoding them, while still being able to successfully decode the messages itself [17]. While the shield can mitigate some of the existing security problems, it offers only very limited protection since adversaries could bypass it by transmitting messages with more power than those sent by the shield. Furthermore, a multiple-input multiple-output (MIMO) eavesdropper could suppress the jamming signal and recover the data sent by the devices, as shown by Tippenhauer et al. [39]. Xu et al. presented the "IMDGuard", a wearable proxy device that performs an authentication process on the ICD's behalf [41]. The IMDGuard relies on patient's electrocardiography (ECG) signals to generate a symmetric key that is valid for one session and is known only to the IMD and itself. However, Rostami et al. found that the IMDGuard is vulnerable to a man-in-the-middle (MiTM) attack which reduces its effective key length from 129 bits to 86 bits [33]. Overall, although external devices that use friendly jamming could help protecting legacy devices, they could also jam transmissions sent by legitimate devices. In some countries, jammers are illegal and their use can result in large fines. Thus, it is unlikely that these solutions will be accepted by the US federal drug administration (FDA) and adopted by manufacturers.

2.2 Out-of-band channels

Another prominent family of countermeasures relies on establishing a cryptographic key between the device programmer and the IMD via an auxiliary or out-of-band (OOB) channel. OOB channels typically have low-bandwidth and are easy to set up. There exist two types of OOB channels: (i) *authentic* or (ii) *secret*. The former represents a channel where Bob is guaranteed that the message he receives was actually sent by Alice. The data can however be eavesdropped by others. The latter represents a channel where Alice is guaranteed that the message she sends is only received by Bob. However, Bob does not know that the data comes from Alice.

Halperin et al. proposed to use a radio-frequency identification (RFID) tag in combination with a piezo-element to achieve audio-based key distribution [20]. They were the first to introduce a countermeasure that does not consume energy from the IMD's battery. Nevertheless, the audio transmissions generated by the piezo-element can be eavesdropped, as shown by Halevi et al. [18]. As this technique uses a carrier frequency within the audible range so that it can be perceived by patients, it is unclear whether this technique could be applied in noisy environments. Kim et al. [24] and Abhishek Anand et al. [10] presented two similar vibration-based key transport protocols through which the device programmer and the IMD can agree on a cryptographic key [24]. Since vibration results in unintentional acoustic emanations, both solutions proposed that the device programmer generates a masking sound to hide the acoustic emanations. However, these solutions require the IMD to have extra hardware, and it is unclear whether the masking sound produced by the device programmer can have an effect in the data sent over the vibration channel. Furthermore, Trippel et al. showed that the integrity of audio and vibration sensor outputs can be compromised by injecting malicious analog acoustic signals [40]. This could allow adversaries to modify the key being sent over the audio/vibration channel in their favor so that the IMD establishes a key with a malicious device. Rasmussen et al. proposed an access control scheme based on ultrasonic distance bounding which enables the IMD to accept any device programmer that is in its close proximity [32]. The main limitation of this solution is that it requires dedicated analog hardware, which makes it not suitable for IMDs. Unfortunately, all the previous solutions have important limitations and drawbacks or have been proven to be vulnerable to security attacks.

The closest solution to ours is the heart-to-heart (H2H) protocol proposed by Rostami et al. [34]. H2H uses a novel access control policy called "touch-to-access" which ensures access to the IMD by any device programmer that can touch the patient's skin to measure his interpulse interval (IPI) (i.e. the time between heart beats). In H2H, both the device programmer and the IMD need to simultaneously measure the patient's heart rate. The heart rate is then used as a "fuzzy password" that is known only to the device programmer and the IMD. The authors state that one of the main advantages of using the patient's heart rate is that it can be measured anywhere in the patient's body just by touching his skin. However, this solution has several weaknesses and limitations. Firstly, Marin et al. showed that

H2H is vulnerable to a reflection and a MiTM attack [26]. Secondly, it is unclear how the authors transformed the IPI signal from analog to its digital form. Thirdly, the H2H protocol has a large communication and computation cost. It requires the IMD to transmit a substantial amount of messages and uses public-key cryptography, which can be too energy consuming for resource-constrained devices such as IMDs. Lastly, several articles have shown that the IPI can be measured remotely using a wide range of techniques. For example, Calleja et al. showed that it is possible to remotely gather information of cardiac signals using widely available inexpensive hardware [13]. Seepers et al. evaluated the feasibility of remote attacks using the existing remote photoplethysmography (rPPG) methods [36]. Their evaluation demonstrates that rPPG achieves similar accuracy as when measuring the heart rate by touching the patient's skin. Poh et al. proposed a simple, low-cost technique through which it is possible to measure the heart rate using a standard webcam [31]. Recently, Katabi et al. demonstrated that WiFi signals can be used to detect the breathing and heart rate of individuals [8]. All these papers render H2H (and other systems relying on the secrecy of the patient's heart rate such as the IMDGuard) insecure. This paper proposes a practical and effective solution that follows the "touch-to-access" principle, but overcomes all the previous limitations.

3 NEUROSTIMULATION SYSTEM

This section describes the devices that comprise the neurostimulation system. This includes device programmers used by doctors and patients as well as neurostimulators.

Device programmers: they are external portable devices used to read out patient's medical data stored on the neurostimulator or to reprogram its settings. There are two types of device programmer: those used by doctors and those used by patients. The former has full privileges to read data and modify the neurostimulator's settings, whereas the latter has restricted permissions, allowing only controlled therapy modifications, as established by the doctor.

Neurostimulators: they are devices implanted subcutaneously near the clavicle that are connected to the brain through several leads. They have limited memory storage, processing power and a battery with limited capacity that cannot be recharged or replaced. When the battery is depleted, the patient needs to undergo surgery in order to get a new implant. Such a surgery always introduces a small, but not negligible risk of infections, sometimes even with lethal consequences. We deliberately chose not to disclose how long the battery lasts because this could implicitly reveal the manufacturer and neurostimulator models that we investigated.

The device programmer contains a magnetic programming head that is used to communicate with the neurostimulator over a wireless bi-directional short-range communication channel (less than 10 cm). The programming head needs to be placed on the patient's chest in close proximity to the neurostimulator for the entire duration of the communication. In most cases, patients have their neurostimulators being interrogated and/or reprogrammed in isolated controlled locations, e.g. the doctor's office. However, patients also need to frequently use their own device programmer in order to adjust the stimulation configuration, for example, when lying, sitting or walking.

4 LABORATORY SETUP

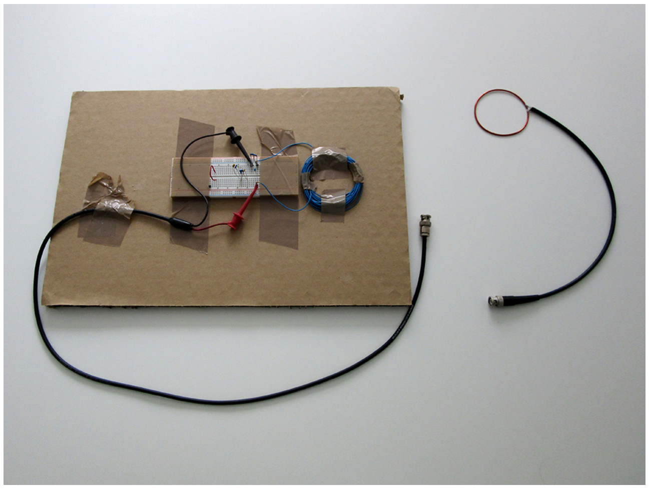

Figure 1: Antennas used for receiving and transmitting signals. The transmit antenna is shown on the left, the receive antenna on the right.

Our laboratory setup comprises inexpensive commercial hardware devices including a standard laptop and a USB-6351 data acquisition system (DAQ) from National Instruments [3]. Moreover, we built two antennas for receiving and transmitting messages respectively, as shown in Fig. 1. For receiving messages, we created a simple antenna by cutting a coaxial cable and connecting a circular piece of copper to it. Even though this antenna allowed us to capture messages exchanged between the devices, its impedance was too low to transmit signals with enough power. To overcome this problem, we designed our own antenna for transmitting signals (for more details, we refer the reader to Appendix A).

5 INTERCEPTING RF TRANSMISSIONS

This paper analyzes the security of the proprietary protocol between the device programmer and the neurostimulator to communicate wirelessly over a short-range communication channel. This is a challenging task because there is no information available about the protocol specifications. IMD manufacturers typically rely on keeping the protocol specifications secret as a means to provide security (i.e. security-through-obscurity). Protocol reverse engineering implies finding both the message format and the protocol state-machine without knowing the protocol specifications. Several articles [20, 27] have already shown that proprietary protocols can be reverse-engineered. While several techniques can be applied to reverse engineer proprietary protocols, physical access to the devices is often necessary, e.g. to extract the firmware.

In this paper, we follow a *black-box reverse engineering* approach which allows us to find the inner-workings of the protocol by only providing inputs to the devices and looking at their outputs. In other words, we change any of the neurostimulator's settings using the device programmer, and then inspect the format of the transmitted messages. Our black-box methodology is a labor-intensive and challenging process yet it is more realistic than other approaches. By following this approach, we assess the feasibility of reverse engineering the proprietary protocol by a weak adversary with limited resources and capabilities who cannot have physical access to the devices but can only intercept the messages sent wirelessly. We acknowledge that having access to the device programmer during our experiments in order to perform certain actions can speed up the process of reverse engineering the protocol. However, although this process may take longer when the actions being performed on the device programmer are not known, adversaries can still learn the inner-working of the protocol by intercepting and analyzing transmissions sent over the air by legitimate devices.

Next, we describe how to reverse engineer the proprietary protocol used by a specific neurostimulator model. However, we also conducted various experiments using other neurostimulator models, and came to similar findings as the ones described in this paper. Figure 3 shows the format of the messages sent by the device programmer.

5.1 Wireless communication parameters

Before capturing the messages exchanged between the device programmer and the neurostimulator, we first needed to discover several wireless communication parameters such as the transmission frequency, modulation and encoding scheme, and the symbol rate being used. This step is crucial as the use of slightly different parameters compared to those used by the devices would result in an incorrect demodulation of the captured messages, i.e. messages with erroneous bits.

The first step of our analysis was to find the frequency at which the devices transmit their messages. By entering the unique device programmer's federal communications commission identifier (FCC ID) in the FCC database [5], we determined that these devices transmit their messages at 175 KHz. We then captured several messages sent by the devices, and visualized the signals in the time domain. The waveform of these signals indicates that an on-off keying (OOK) modulation scheme is being used, as illustrated in Fig. 2. In an OOK modulation, the presence of a carrier wave is used to indicate a binary '1' and its absence indicates a binary '0'. Similarly to passive RFID devices, all messages are encoded to ensure that the neurostimulator receives enough power from the device programmer even when a long string of "0s" is sent. Encoding of '1' or '0' is based on a variable width high-pulse and a fixed width space. For security reasons, we decided not to reveal the symbol rate being used.

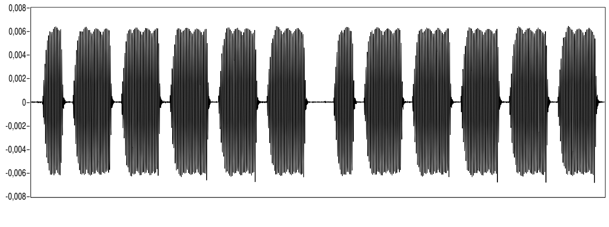

Figure 2: Waveform of a signal transmitted by the device programmer.

5.2 Reverse engineering the proprietary protocol

We intercepted messages exchanged between the two types of device programmer and several models of neurostimulator. However, we focused on analyzing the messages sent by the device programmer since understanding these transmissions would allow us to emulate a legitimate device programmer. (In Sect. 5.3 we show that most messages sent by neurostimulators are simply acknowledgements). After demodulating the captured messages, we observed that they all include a common synchronization sequence (not shown in Fig. 3) that consists of a series of alternating '1s' and '0s'. Subsequently, we found several message fields including headers, information data and checksums.

As a first experiment, we grouped the messages sent by the doctor's device programmers and patient's device programmers, respectively, in two different clusters. This experiment allowed us to determine that there is a 16-bit sequence at the beginning of each message which can take two values depending on the type of device programmer being used. Similarly, we compared the messages sent by each different device programmer, and discovered that there is a unique 16-bit sequence that varies depending on the device. This led us to conclude that this field represents the unique serial number (SN) of each device programmer. Following this approach, we also found two 16-bit sequences that denote the model and SN of the neurostimulator, respectively.

These headers are followed by a static x-bit sequence that is used to distinguish between three states of the communication including (i) unknown neurostimulator's SN, (ii) known neurostimulator's SN and (iii) confirmed neurostimulator's SN. More specifically, a sequence is used in the first message sent by the device programmer to indicate that the neurostimulator's SN is not yet known. A different sequence is used in the second message sent by the device programmer to indicate that both devices already know each other. From that point onwards, the device programmer uses a third sequence until the communication session finishes or expires. Subsequently, we intercepted the messages sent from the device programmer to the neurostimulator while performing different actions. By analyzing these messages, we observed that each message has a y-bit field that corresponds to the action being conducted by the device programmer, e.g. changing the patient's information or the therapy settings. Next, there is a payload field that contains different information depending on the settings being reprogrammed. Within the payload field, the data is encoded in ASCII and transmitted in reversed order, i.e. the least significant bit becomes the most significant bit and vice versa.

At the end of some messages, there are two 16-bit sequences that seem to be uniformly distributed. This made us think that they could be checksums to detect or correct bit errors. Our first hypothesis was that these sequences correspond to a cyclic redundancy check (CRC). This is because CRCs are widely used, easy to implement and very effective at detecting bit errors. Since CRCs are linear functions, we first checked whether the linearity property holds for the second 16-bit checksum. For this purpose, we computed the XOR of two messages and verified whether the result produces a valid message. As the linearity property was satisfied, the second step was to find

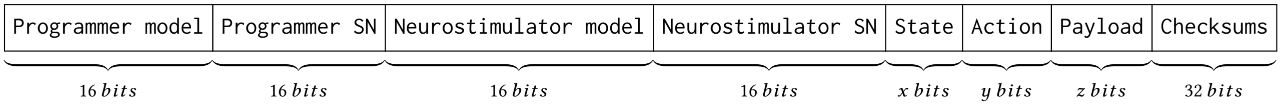

Figure 3: Device programmer's message format.

all the CRC parameters such as the polynomial, initial XOR value, final XOR value as well as whether the input and/or the output is reflected. To discover these parameters, we created a program that brute-forces all possible combinations of CRC parameters. This task required less than 5 minutes on a standard laptop. Following the steps described above, we discovered that the second 16-bit checksum corresponds to a CRC checksum that is computed over the entire message using the standard CRC-16-CCITT [25].

We then repeated this approach for the first 16-bit checksum but we did not succeed on finding the CRC parameters. A key observation was that this checksum remains identical when performing a specific action regardless of the device programmer and/or neurostimulators being used. In other words, this checksum depends only on the state, action and payload fields. To create messages with a valid checksum, we intercepted several messages sent by the device programmer, and XORed them to create new messages where only one bit is set to '1' and the rest of bits are set to '0'. We repeated this approach for each of the bits within these message fields, and created a code-book with all possible messages and their corresponding checksums. Let us give an example to describe how to compute the first 16-bit checksum of a message using our approach. Assuming that our message contains a '1' in the third, fifth and eleventh position, then we can compute this checksum by XORing the checksums of the third, fifth and eleventh messages, respectively, within our code-book.

5.3 Protocol state-machine

This section describes the three phases of the communication between the device programmer and the neurostimulator. This includes the initialization phase, the reprogramming phase and the termination phase.

Initialization phase: Initially, both devices exchange several messages so that the device programmer requests all the information stored on the neurostimulator.

We found that the device programmer always starts the communication by sending a message that is identical across sessions which contains the model and serial number of the device programmer. Within this message, the fields that denote the SN and model of the neurostimulator are kept empty. This is because the device programmer is not linked to any neurostimulator and does not yet know with which neurostimulator it is communicating. The neurostimulator then replies with a message containing its SN and model. From that point until the end of this phase, the device programmer always sends one message to request data whereas the neurostimulator replies back with two distinct messages. The former is an acknowledgment that indicates whether the message was received correctly, while the latter contains the data.

Reprogramming phase: After the initialization phase, the device programmer can be used to modify the neurostimulator's settings as many times as needed within the same protocol session. We discovered that the device programmer sends only one message to adjust the settings, to which the neurostimulator replies with two different messages that remain identical regardless of the action being performed.

Termination phase: Before an ongoing communication is terminated, the device programmer sends a message to the neurostimulator which enables the latter to switch to power saving mode. The neurostimulator replies back with the same two messages that it sends after it has been reprogrammed.

In the next section, we show that it is not necessary to follow the normal protocol flow in order to perform attacks on the neurostimulators.

6 SOFTWARE RADIO-BASED ATTACKS

This section details several software radio-based attacks that we are able to perform on the neurostimulator which could endanger the safety and compromise the privacy of patients. We conducted all these attacks with the device programmer turned off. We focus on an experiment where we changed the patient's name (programmed on the neurostimulator) since this allows us to easily verify whether the experiment succeeded. However, we want to emphasize that we can mimic the behavior of a legitimate device programmer to send valid messages to a neurostimulator to change any of its settings.

Replay attacks: We were able to modify any of the neurostimulator settings just by intercepting and replaying past transmissions sent from legitimate device programmers. However, this attack has two important practical limitations. The adversary needs to wait until there is an ongoing communication between a device programmer and a neurostimulator in order to intercept the messages. Furthermore, the adversary can only replay messages that were already transmitted by legitimate device programmers, which clearly limits the impact of the attack.

Spoofing attacks: Unlike the previous attack, spoofing attacks require to have partial or full knowledge of the protocol. After reverse engineering the protocol, we were able to create any arbitrary message and send it to the neurostimulator. One possible limitation that makes this attack less practical is that the adversary needs to know the neurostimulator's SN in order to create a valid message. Intuitively, adversaries could obtain the neurostimulator's SN by intercepting the exchanged messages while there is an ongoing communication. As the first message sent by the device programmer is always identical regardless of the neurostimulator, adversaries could also replay this message to a neurostimulator and intercept the response since this contains its SN.

We found a way to overcome even this limitation, which allows adversaries to send messages to a neurostimulator without knowing its SN. More specifically, we discovered that neurostimulators accept messages with an empty neurostimulator SN field. In our experiments, we were able to change the patient's name, programmed in the neurostimulator, by sending a single message with no neurostimulator's SN. The impact of this attack is quite large as an adversary could create a valid message, without a SN, and reuse it for all neurostimulators. The only challenge for adversaries is to be close enough to the victim in order to communicate to the neurostimulator. However, there are definitely several scenarios, such as a crowded subway, where this would be possible.

Privacy attacks: Since our reverse engineering on the proprietary protocol shows that the data sent over the air is unencrypted, passive adversaries can eavesdrop the wireless channel to infer private information about patients. Active adversaries can additionally send messages to the neurostimulator to request specific data. The data sent over the air between the device programmer and the neurostimulator includes diagnosis, symptoms, disease or therapy information. Moreover, all messages exchanged between the devices include a unique neurostimulator SN which could be used for adversaries to identify, locate and track patients. For this purpose, adversaries could install beacons in strategic locations (e.g. in hospitals or train stations) to learn the patients' movement pattern based on the signals transmitted by their neurostimulators.

DoS attacks: One would expect that neurostimulators switch to "sleep mode" once an ongoing communication is terminated. However, we found that neurostimulators always remain active and accept a message as long as its format and checksums are correct. We also observed that adversaries can send valid messages to the neurostimulator without needing to first execute the initialization phase. While we did not try to quantify the effect of performing a DoS attack against these devices, adversaries could repeatedly send malicious messages to the neurostimulator in order to deplete its resources faster, similarly to the attacks proposed by Hei et al. [21].

7 SECURITY ARCHITECTURE

In this paper, we propose a practical security architecture that allows the device programmer and the neurostimulator to securely exchange messages. Our solution consists of three main items: (i) session key initialization, (ii) key transport and (iii) secure data communication. The first two items are particularly challenging for medical implants.

To bootstrap a secure communication channel, a symmetric (session) key need to be shared between the devices. This key could be generated by the device programmer and transported from the device programmer to the neurostimulator using an OOB channel. However, this approach presents two limitations. Most OOB channels that allow to transport a key from the device programmer to the IMD, are shown to be vulnerable to security attacks or require to add extra hardware components in the neurostimulator. An alternative solution would be to generate the key in the neurostimulator and transport it to the device programmer. Halperin et al. presented a audio-based key transport solution where the IMD generates a random symmetric key and transports it to the device programmer through an acoustic channel [20]. Even though the authors stated that the key can be received only by a device programmer that can make physical contact with the patient, Halevi et al. showed that it is possible to recover the key from far away by eavesdropping the acoustic channel [18].

We propose a solution where a symmetric (session) key is also generated in the IMD and transported to the device programmer using a secret OOB. Our solution overcomes the previous limitations and ensures that the key can only be picked up by a device programmer that touches the patient's skin for a few seconds. Below, we define our adversarial model and give an overview of all the building blocks of our solution.

7.1 Adversarial model

We consider the presence of strong adversaries who can eavesdrop or jam the wireless channel, as well as modify, replay or forge messages. Adversaries can be in close proximity with the patient and can possess any legitimate device programmer, even those that already interacted with the neurostimulator. However, adversaries cannot compromise the neurostimulator or the device programmer while being used, since this would make it impossible to protect the neurostimulator. Doctors are trusted and do not collude with adversaries. Adversaries can neither touch the patient's skin long enough (i.e. few seconds) without the patient noticing it, nor compromise any device that can make physical contact to the patient (e.g. a smart watch). Within the community, it is accepted to assume that physical contact to a patient means the ability to cure or harm [34]. When designing security solutions and defining the adversarial model for IMDs, it is important to note that their primary goal is to treat patients. Safety is of utmost importance and security should never interfere with this task.

7.2 Overview of the solution

Our solution involves three main steps: (i) key generation, (ii) key transport and (iii) secure data exchange. More concretely, our solution works as follows. Firstly, the neurostimulator uses a physiological signal from the patient's brain as a source of randomness for generating a 128-bit symmetric key. This key is valid only for a single session and is independent from old and future keys. Thus, if an adversary ever compromises a session key, he will not be able to compute past and future keys, i.e. backward and forward security are guaranteed. Secondly, this session key is transported securely and reliably from the neurostimulator to the device programmer through a secret OOB channel. Similarly to H2H [34], our solution follows a "touch-to-access" access control policy where access to the neurostimulator is granted only to device programmers that can touch the patient's skin for a few seconds. This provides a practical yet effective balance between security and permissive access in emergencies since it allows medical personnel to have immediate access to the neurostimulator without needing to contact care providers for device-specific cryptographic keys. In contrast to the existing solutions, key transportation can be achieved without adding extra hardware components on the neurostimulator. Once the key has been transported, the devices can optionally run a standard authentication protocol so that they can prove to each other that they posses the session key. Finally, this session key is used to bootstrap a secure communication channel between the devices.

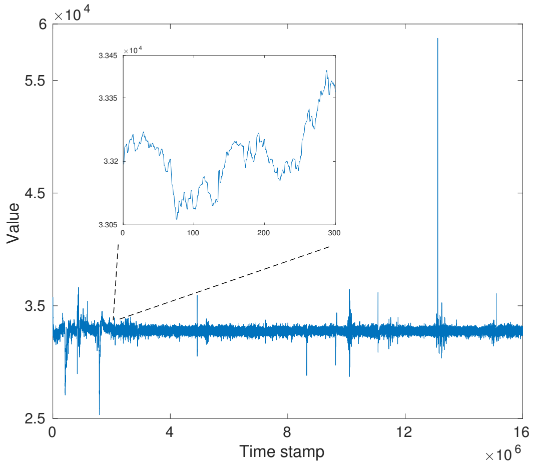

Figure 4: LFP data of one of our mice collected from one channel and sampled with 16-bit precision.

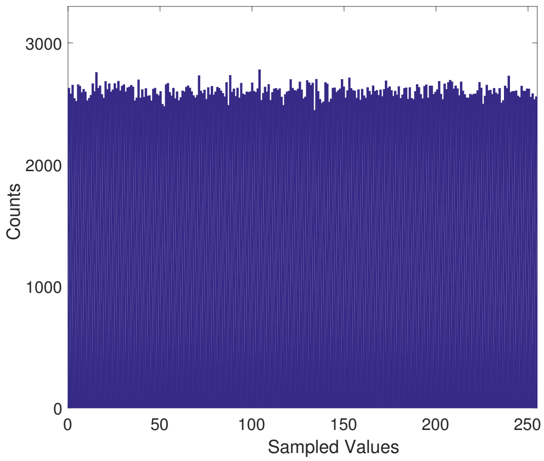

Figure 5: Histogram of bits (in groups of 8 consecutive bits) after applying the parity filter.

7.3 Key generation

True random number generators (TRNGs) are an essential building block of cryptographic systems for generating cryptographic keys, nonces and masks. Unfortunately, IMDs typically use microcontroller-based platforms and lack dedicated TRNGs, making it non-trivial to generate random numbers on these devices. Several articles have proposed to use the initial content of the static random-access memory (SRAM), on-chip RC oscillators and on-board external clocks as an alternative to TRNGs [22, 23]. However, these approaches need to be evaluated for each specific device since their performance depends on the platform and hardware components being used.

Recently, the use of physiological signals extracted from the patient's body has been proposed for generating cryptographic keys. This approach has several advantages with respect to traditional approaches based on PRNGs or TRNGs. Specifically, they allow to reuse signals that are already being gathered by the devices as a low-cost source of randomness. A possible limitation is that some physiological signals, such as those extracted from the patient's heart, can be predicted or measured remotely, which makes them vulnerable to attacks. In this paper, we explore the potential of using a signal from the patient's brain, that cannot be measured remotely by adversaries, as a randomness source for generating cryptographic keys. Our approach can be applied to any IMD that can measure the LFP signal from the patient's brain.

7.3.1 ***LFP signal as a randomness source.*** We propose to use a physiological signal from the patient's brain called *local field potential* (LFP), which refers to the electric potential in the extracellular space around neurons. The choice of the LFP for randomness extraction is based on the following reasons. Firstly, neurostimulators can easily measure signals in the patient's brain. There are already neurostimulators on the market that use the LFP to create feedback for the delivered stimulation. As the LFP can be collected with the existing lead configurations (i.e. without changing the leads' position or requiring extra leads), future generations of neurostimulators will require only minor software changes to be able to measure this signal. Secondly, unlike other physiological signals such as the ECG, the LFP can be measured only through direct contact with the patient's brain, thus cannot be obtained remotely.

To analyze the feasibility of extracting randomness from the LFP to generate a 128-bit key in the neurostimulator, we used real LFP data collected from 22 mice[1]. Figure 4 shows LFP data collected from one of the 22 mice. While we recorded LFP data simultaneously from 16 electrode contacts connected to the mice brain using the W16 wireless recording device [7], we only looked at one of these channels. Our technique to extract randomness can be applied to LFP data collected from any of the 16 different channels. During each LFP recording, the mice were walking on a horizontal ladder. This is a well-known test that is used for many different purposes (for more details about this test, see Metz and Wishaw [30]).

The first step in our assessment was to extract the least significant bit of the sampled LFP data and used it as the raw random number. This is because the lower bits of LFP data seem to be quite noisy. Another possibility would be to extract multiple bits from the sampled LFP data but this would require to evaluate their joint probability distribution. We then computed the XOR of three consecutive bits using a simple two-stage parity filter to increase the entropy density. Figure 5 shows the histogram of 8 consecutive bits after applying the two-stage parity filter. Based on our results, these bits are to some extend uniformly distributed, which demonstrate the potential of the LFP as a randomness source for deriving a symmetric key. Following the test suites from NIST 800-90B [12], we estimated that the Shannon-entropy is around $0.91/bit$. The Shannon-entropy could be further improved by increasing the number of stages of the parity filter.

[1] Using data from mice to extract preliminary results to better understand the human brain is common practice in many scientific disciplines.

7.4 Key transport

Once a fresh cryptographic session key is generated in the neurostimulator, it has to be securely transported to the device programmer. Our technique for key transportation leverages the fact that both the patient's skin and the neurostimulator's case are conductive. We propose to apply an electrical signal (with the key bits embedded in it) to the neurostimulator's case such that the device programmer can measure this signal by touching the patient's skin, as shown in Fig. 6. To realize this technique, two extra short wires are needed from the microcontroller to the case of the neurostimulator. We discussed this technique with doctors and they claim that applying a signal of a few microvolts or millivolts to the neurostimulator's case would not cause any problem or be unpleasant to patients.

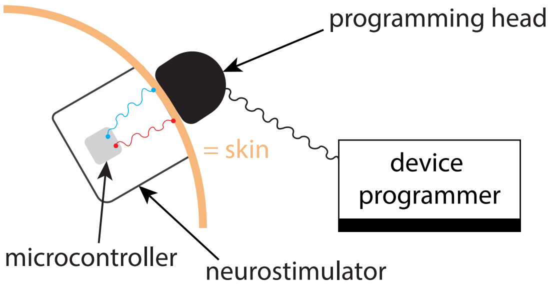

Figure 6: Technique to transport a session key from the neurostimulator to the device programmer.

To evaluate the suitability of this technique, we implemented a proof-of-concept using a NI USB-6351 DAQ and a neurostimulator. To emulate the human body, we used a 1 cm layer of bacon on a 4 cm layer of beef, as suggested by Kim et al. [24]. While this model does not account for changes in the skin conductance, for example, due to patient's emotions or sweat, all these factors can only make the patient's skin to be slightly more conductive than usual. In other words, none of these factors will affect the reliability of our technique. However, this could make it easier for adversaries to capture the EM radiations generated when transporting the key.

To preclude potential EM eavesdropping attacks, our system should be designed to minimize undesired EM radiation produced by the hardware components. This can be done by lowering the transmission power. While this comes at the cost of having a lower data rate, this would not be a problem in our solution. Another way of decreasing the EM emanations would be to use wires (from the microcontroller to the neurostimulator's case) that are sufficiently short and twisted. To further decrease the EM emanations, one could follow widely used electromagnetic compatibility (EMC) guidelines [4]. In addition, the undesired EM emanations generated from other devices that are in close proximity with the patient (e.g. his smart-phone) could be used to masquerade the EM radiations.

To simulate the process of transporting the key, we created a transmitter and a receiver LabVIEW program. We modulated the data using a standard OOK and set the symbol rate and the sampling rate to 100 symbols/s and 500 ksamples/s, respectively. Given that the duration of each symbol is 10 ms, the 128-bit key can be transported in less than 1.5 s. A few additional delays could be deliberately introduced in the key transport process such that access to the neurostimulator is guaranteed only to device programmers that can touch the patient's skin for a few seconds.

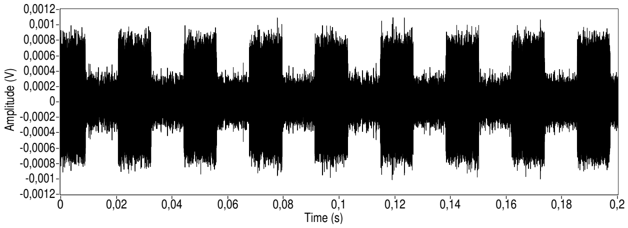

Figure 7: Waveform of the signal received by our DAQ at the other side of the meat.

For our experiments, we connected two wires from the transmission port of the DAQ to the neurostimulator's case. We then placed the meat on top of the neurostimulator, and attempted to measure the transmitted signal at the other side of the meat. Then we mimicked the behavior of a legitimate device programmer by using two wires to touch the meat while connected to the receiver port of the DAQ. The first step was to modulate the "1s" and "0s" corresponding to the 128-bit session key using the OOK modulator. Subsequently, we applied the modulated electrical signal to the neurostimulator's case using our transmitter. A short preamble sequence could be sent before the key is transmitted to help to synchronize the devices. Existing techniques to detect and correct bit errors could also be used to increase the robustness of our technique. Using our receiver, we measured and demodulated the signal to retrieve the key bits previously transmitted by the neurostimulator.

We conducted a series of experiments to determine the minimum signal power from which the device programmer can recover the key bits transmitted by the neurostimulator. It is important to note that the transmission power could be even further reduced by using an enhanced multi-feature OOK demodulation scheme, as the one proposed by Kim et al. [24]. Figure 7 illustrates the waveform of the signal (with the key bits embedded in it) that is received by our DAQ at the other side of the meat. This figure shows that it is possible to recover the key bits, which consists of a series of alternating '1s' and '0s', by demodulating a signal whose amplitude is less than 1 mV. Our results show that, when using the parameters mentioned above, the key can successfully be transported from the neurostimulator to the device programmer.

Furthermore, we tested whether it is possible to recover the key without needing to touch the patient's body when using the minimum signal power that we used in the experiment discussed above. For this, we connected two wires to the receiver port of the DAQ and used them as an antenna to try to measure the EM emanations. We repeated this experiment from several distances ranging from 3 meters to a few centimeters. In none of these cases, we were able to capture the transmitted key. This led us to conclude that the key can only be successfully received when touching the patient's skin.

Programmer model	Programmer SN	Neurostimulator model	Neurostimulator SN	Counter	State	Action	Payload	MAC
16 *bits*	16 *bits*	16 *bits*	16 *bits*	12 *bits*	*x bits*	*y bits*	*z bits*	64 *bits*

Figure 8: Optimized message format.

7.5 Secure data exchange

For the sake of completeness, we describe how to establish a secure communication channel once the session key has been transported from the neurostimulator to the device programmer. Firstly, both devices could optionally execute any authentication protocol to prove to each other knowledge of the shared session key. For efficiency reasons, we chose not to use an authentication protocol. Instead, both devices can implicitly demonstrate knowledge of the key during the first communication session.

The use of cryptography allows for secure data exchange between devices. However, it also increases the energy consumption in both the neurostimulator and the device programmer. This energy consumption can be divided into two components: (i) computation and (ii) communication cost. The former indicates the cost of performing cryptographic operations while the latter refers to the cost of transmitting/receiving bits to/from a device. However, several papers have shown that the computation cost is often negligible compared to the communication cost [28, 37]. Thus, we propose to use a new optimized message format that is slightly different from the original message format (see Fig. 3). This allows to build security mechanisms into the devices while keeping the additional energy consumption as low as possible in the neurostimulator.

Figure 8 shows the new optimized message format. It includes the same fields as those in the original message except for the two 16-bit checksums, and additionally has a 12-bit counter and a 64-bit MAC that is computed over the entire message. The reasoning behind the proposed message format optimization is the following. A 64-bit MAC offers a good trade-off between cost and security against both off-line and on-line attacks. To prevent replay attacks, the 12-bit counter is increased by one in each message and reset every time a new session key is generated. A message is accepted only if its counter is greater than the one in the previous received message. All these message optimizations lead to an increase of the message size of only 44 bits, which corresponds to an extra communication overhead of less than 10% compared to the original message format.

Since we need to encrypt all message fields except for the SN and model fields of both the device programmer and the neurostimulator, we recommend to use any secure authenticated encryption scheme [2].

8 LIMITATIONS AND FUTURE WORK

Lack of stochastic model for LFP. Our results indicate that the use of the LFP signal is a promising way for extracting randomness in neurostimulators. However, we conducted all our experiments without having a stochastic model on the entropy source or a physical model on the mechanism of the signal origin. In future work, we will develop a stochastic model for the LFP, and conduct a thorough analysis to gather more evidence that LFP can be used as a source of randomness in neurostimulators.

Consider a more powerful adversarial model. Our current solution assumes that adversaries cannot compromise any device that can make physical contact to the patient (e.g. a smart watch). In future work, we want to relax this requirement and also allow the adversary to compromise any wearable device of the patient.

9 CONCLUSIONS

In this work we have evaluated the security and privacy properties of a widely used commercial neurostimulator. For this, we fully reverse engineered the proprietary protocol between the device programmer and the neurostimulator over a short-range communication channel. We demonstrated that reverse engineering was possible without needing to have physical access to the devices by using a black-box approach. This allowed us not only to document the message format and the protocol state-machine, but also to discover that the messages exchanged between the devices are neither encrypted nor authenticated. We conducted several software radio-based attacks that could endanger the patients' safety or compromise their privacy, and showed that these attacks can be performed using inexpensive hardware devices. The main lesson to be learned is that security-through-obscurity is always a dangerous design approach that often conceals insecure designs. IMD manufacturers should migrate from weak closed proprietary solutions to open and thoroughly evaluated security solutions and use them according to the guidelines.

To preclude the above attacks, we presented a practical and complete security architecture through which the device programmer and the neurostimulator can agree on a session key that allows to bootstrap a secure communication channel. Our solution grants access to the neurostimulator to any device programmer that can touch the patient's skin for a few seconds. This allows to create a secure data exchange between devices while ensuring that medical personnel can have immediate access to the neurostimulator in emergencies. Our solution accounts for the unique constraints and functional requirements of IMDs, requires only minor hardware changes in the devices and provides backward and forward security.

10 ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their helpful comments and Stefanos Georgoutsos for his support. This work was supported in part by the Research Council KU Leuven: C16/15/058 and by FWO through SBO SPITE S002417N.

REFERENCES

[1] American Academy of Pain Medicine. http://www.painmed.org/patientcenter/facts_on_pain.aspx. [Online; accessed 5-June-2017].

[2] CAESAR competition. https://competitions.cr.yp.to/caesar.html. [Online; accessed 5-June-2017].

[3] DAQ NI USB-6351. http://www.ni.com/en-us/support/model.usb-6351.html. [Online; accessed 5-June-2017].

[4] EMC Improvement Guidelines. http://www.atmel.com/images/doc4279.pdf. [Online; accessed 5-June-2017].

[5] Federal Communications Commission (FCC) ID. http://www.fcc.gov/encyclopedia/fcc-search-tools. [Online; accessed 5-June-2017].

[6] Parkinson association. http://www.parkinsonassociation.org/facts-about-parkinsons-disease/. [Online; accessed 5-June-2017].

[7] W16 wireless recording device. http://www.multichannelsystems.com/products/wireless-systems. [Online; accessed 5-June-2017].

[8] WiFi system detects peoples breathing heart rate even through walls. https://www.medgadget.com/2014/06/mits-wifi-system-detects-peoples-breathing-heart-rate-even-through-walls.html. [Online; accessed 5-June-2017].

[9] C. Adams and S. Lloyd. *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.

[10] S. A. Anand and N. Saxena. Vibreaker: Securing Vibrational Pairing with Deliberate Acoustic Noise. In *Proceedings of the 9th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec, pages 103–108, NY, USA, 2016. ACM.

[11] C. A. Balanis. *Antenna Theory: Analysis and Design*. Wiley-Interscience, 2005.

[12] E. Barker and J. Kelsey. Recommendation for the entropy sources used for random bitgeneration. NIST DRAFT Special Publication 800-90B, 2016.

[13] A. Calleja, P. Peris-Lopez, and J. E. Tapiador. *Electrical Heart Signals can be Monitored from the Moon: Security Implications for IPI-Based Protocols*, pages 36–51. Springer International Publishing, Cham, 2015.

[14] L. Chunxiao, A. Raghunathan, and N. Jha. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In *13th Int. Conf. on e-Health Networking Applications and Services*, pages 150–156, Jun 2011.

[15] T. Denning, Y. Matsuoka, and T. Kohno. Neurosecurity: security and privacy for neural devices. *Neurosurgical Focus*, 27(1):E7, 2009.

[16] C. F. *Capacitor tuning circuits for inductive loads*. PPG-1401, 1992.

[17] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu. They Can Hear Your Heartbeats: Non-invasive Security for Implantable Medical Devices. *SIGCOMM Comput. Commun. Rev.*, 41(4):2–13, Aug. 2011.

[18] T. Halevi and N. Saxena. On pairing constrained wireless devices based on secrecy of auxiliary channels: the case of acoustic eavesdropping. In *Proc. 17th Conf. on Computer and Communications Security, October 4-8*, pages 97–108, 2010.

[19] D. Halperin, T. S. Heydt-Benjamin, K. Fu, T. Kohno, and W. H. Maisel. Security and Privacy for Implantable Medical Devices. *IEEE Pervasive Computing, Special Issue on Implantable Electronics*, 7(1):30–39, Jan. 2008.

[20] 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 *Proc. 29th IEEE Symposium on Security and Privacy*, pages 129–142, May 2008.

[21] X. Hei, X. Du, J. Wu, and F. Hu. Defending resource depletion attacks on implantable medical devices. In *Global Telecommunications Conference*, pages 1–5, Dec 2010.

[22] J. Hlavác and R. Ló. True random number generation on an Atmel AVR microcontroller. In *2nd Int. Conf. on Computer Engineering and Technology*, volume 2, pages V2–493–V2–495, April 2010.

[23] D. E. Holcomb, W. P. Burleson, and K. Fu. Power-up sram state as an identifying fingerprint and source of true random numbers. *IEEE Trans. Comput.*, 58(9):1198–1210, Sept. 2009.

[24] 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*, DAC '15, pages 32:1–32:6, New York, NY, USA, 2015. ACM.

[25] P. Koopman and T. Chakravarty. Cyclic redundancy code (CRC) polynomial selection for embedded networks. In *Int. Conf. Dependable Systems and Networks*, pages 145–154, June 2004.

[26] E. Marin, E. Argones Rúa, D. Singelée, and B. Preneel. A survey on physiological-signal-based security for medical devices. Cryptology ePrint Archive, Report 2016/188, 2016. http://eprint.iacr.org/.

[27] 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 *Proc. 32th Annual Conference on Computer Security Applications*, pages 226–236, 2016.

[28] E. Marin, D. Singelée, B. Yang, I. Verbauwhede, and B. Preneel. On the feasibility of cryptography for a wireless insulin pump system. In *Proc. 6th Conf. on Data and Application Security and Privacy*, pages 113–120, 2016.

[29] I. Martinovic, D. Davies, M. Frank, D. Perito, T. Ros, and D. Song. On the feasibility of side-channel attacks with brain-computer interfaces. In *Proc. 21st USENIX Conference on Security Symposium*, pages 34–34, 2012.

[30] G. A. Metz and I. Q. Whishaw. The ladder rung walking task: a scoring system and its practical application. *Journal of Visualized Experiments*, (28):e1204–e1204, 2009.

[31] M. Z. Poh, D. J. McDuff, and R. W. Picard. Advancements in Noncontact, Multiparameter Physiological Measurements Using a Webcam. *IEEE Transactions on Biomedical Engineering*, 58(1):7–11, Jan 2011.

[32] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun. Proximity-based access control for implantable medical devices. In *Proc. 16th Conf. on Computer and Communications Security*, pages 410–419, 2009.

[33] M. Rostami, W. Burleson, F. Koushanfar, and A. Juels. Balancing security and utility in medical devices? In *50th Design Automation Conference, May 29 - June 07*, pages 13:1–13:6, 2013.

[34] M. Rostami, A. Juels, and F. Koushanfar. Heart-to-heart (H2H): authentication for implanted medical devices. In *Conf. on Computer and Communications Security, November 4-8* [34], pages 1099–1112.

[35] 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 *Proc. IEEE Symposium on Security and Privacy*, pages 524–539, May 2014.

[36] R. M. Seepers, W. Wang, G. de Haan, I. Sourdis, and C. Strydis. Attacks on heartbeat-based security using remote photoplethysmography. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2017.

[37] D. Singelee, S. Seys, L. Batina, and I. Verbauwhede. The communication and computation cost of wireless security: Extended abstract. In *Proceedings of the Fourth ACM Conference on Wireless Network Security*, WiSec '11, pages 1–4, New York, NY, USA, 2011. ACM.

[38] F. Stajano and R. J. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Proc. 7th Int. Workshop on Security Protocols*, pages 172–194. Springer-Verlag, 2000.

[39] N. O. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun. On Limitations of Friendly Jamming for Confidentiality. In *Proc. IEEE Symposium on Security and Privacy*, pages 160–173, May 2013.

[40] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu. WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks. In *Proc. 2nd European Symposium on Security and Privacy*, Apr. 2017.

[41] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li. IMDGuard: Securing implantable medical devices with the external wearable guardian. In *2011. 30th Int. Conf. on Computer Communications, 10-15 April*, pages 1862–1870, 2011.

A DESIGN OF AN ANTENNA OPERATING AT 175 KHZ

Since there are no off-the-shelf antennas suitable for transmitting at 175 kHz, this component had to be designed and manufactured in house. The low working frequency points at a loop antenna as the optimal solution. A small loop antenna with N turns and a surface area S carrying an electric current I_o behaves similarly to a magnetic dipole with magnetic moment $I_m l$ given in Eqn. 1:

$$I_m l \approx NSI_o \, . \tag{1}$$

Thus, to maximize the magnetic field component, the equivalent dipole moment should be maximized. The conventional topology for near field communication systems assumes that the antenna size is comparable with the size of the neurostimulator antenna. Thus, the surface area S is determined by the size of the neurostimulator. The number of turns N can easily be adjusted. The electric current I_o depends on the output power and on the antenna matching. A small loop antenna behaves like an inductor with very small losses [11]. So it is not matched with a conventional 50 Ohm output and special antenna tuning is required. Due to the low frequency this tuning circuit can be constructed using lumped elements. The simplest type of matching network is a so called L matching network based on two reactive elements to match almost any load impedance to a 50 Ohm output at a single frequency [16].

Unfortunately, due to the large detuning of the antenna, the antenna impedance and the exact circuit model of the lumped elements, thus including parasitic components, should be known at the working frequency with a very high accuracy. Because the capacitor equivalent series resistance (ESR) can be comparable with the very small antenna resistance, the design of such a small loop antenna is typically based on a trial and error approach. We manufactured the loop antenna from a 4 m long blue 7 x 0.25 mm cable with PVC insulation from LappKabel. The antenna input impedance was measured using a low cost vector analyzer called miniVNA. The measured value was about 1.8 + j 79.4 Îľ at 175 kHz. The input antenna resistance is determined mainly by the Ohmic wire resistance. A perfect matching can be obtained with two ideal capacitors of 2.4 nF and 9.4 nF. Since there were no such capacitors available, the actual tuning was performed by the consecutive testing of different capacitors. The final matching circuit contains two capacitors of 1 nF in parallel and 1 capacitor of 10 nF. All capacitors were fixed on a breadboard to ensure the possibility to easily re-tune if necessary. The magnetic field level was tested using a 6 cm H-field probe, Model 901 from the EMCO 7405 Near Field probe kit, and an Anritsu MS2721A spectrum analyzer. At first the field level of an original RF transmission was measured and recorded. Then the recorded message was re-transmitted using the set-up based on our antenna, and the magnetic field level was again measured. The new transmitter with the designed antenna provides a magnetic field level and a bandwidth comparable with those generated by the original neurostimulator.