# An Authentication Protocol for Implantable Medical Devices

by

Michal Loin

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday February 12, 2019 at 1:30 PM.

| | | |
|---|---|---|
| Student number: | 4587324 | |
| Project duration: | January 1, 2018 – February 12, 2020 | |
| Thesis committee: | Dr. C. Doerr, | TU Delft, supervisor |
| | Prof. dr. ir. J. C. A. van der Lubbe, | TU Delft |
| | Dr. C. Strydis, | Erasmus Medical Center |

*This thesis is confidential and cannot be made public until February 12, 2020.*



TU Delft

# Abstract

Proper security mechanism are a crucial part of safe usage of Implantable Medical Devices. Multiple researchers presented various solutions to address this problem, basing them on different underlying principles. Within the scope of this thesis we perform a security analysis of the chosen authentication protocols. What is more, we present a new attack on a scheme based on physiological signal processing using a fuzzy vault cryptographic primitive. We exploit the fact that the signal generated by the heart beats does not change sufficiently in the frequency domain. Therefore it is possible that the adversary reuses signal recorded at some earlier point of time to authenticate to the implant in real time. We show in an experimental way that it is able to break the scheme with probability reaching 75%. Finally, we propose a novel lightweight authentication protocol based on hash chains. To ensure the applicability of our work, we have decided to use only energy efficient solutions, that is hash functions and block ciphers. In contrast to existing work, we have extended the threat model and considered the implant reader distrusted. We present a set of energy measurements to provide advantages of different elements to be used during implementation of our solution.

# Contents

# 1 Introduction

Implantable Medical Devices (IMDs) are an important addition to the modern medicine. Ranging from pacemakers to neurostimulators, they help doctors to monitor and treat different diseases, such as cardiac conditions or epileptic seizures. As the name suggests, these devices are fully implanted in the human body, and are able to work independently for around 10 years [47]. Because of that, they are heavily resource constrained devices, with small size limiting battery capacity. When the battery is drained, the patients need to undergo a surgery, where the whole implant is replaced. Therefore, all of the performed computations must be as energy efficient as possible. Throughout the years of development, IMDs have been changing to further improve quality of life of patients. One of the important additions was the introduction of radio transceiver, enabling wireless communication with external devices. Through this channel, doctors could obtain medical measurements of patient's body, as well as adjust the healing program of the device, without a need of a surgery. Nowadays, a typical set up involving an IMD includes an external reader, capable of communicating with the implant.

## 1.1. Importance of security for medical devices

The introduction of wireless communication modules to the IMDs created new security and privacy issues. In general, we can classify the threats based on the goals they try to achieve. The first basic property that can be violated is the *authenticity*. Before any operation is performed, the identities of involved parties must be established. In the impersonation attack, the adversary can impersonate either the IMD or the programmer if the wireless channel is not properly protected [2]. By imitating the programmer, adversary can get access to the implant, and harvest sensitive information about the patient, or in the worst case scenario reprogram the device. By impersonating the implant, the adversary can send false data to the physician. This can desynchronize the real state of the implant and state known by the doctors. It could cause delays in the responses to the needs of the patient, which is a direct life threat. In the scenarios different than IMD, authentication can be achieved relatively easily, with usage of shared keys or infrastructure of public key certificates being the prime examples. However, various programmers must be able to pair with different implants, which could cause large amount of data stored on the constrained devices. Additionally, compromising the programmer and stealing the keys is easier than performing similar attacks on Internet services. There exists also another

aspect of security, similar to the authentication - *authorization*. Authorization of (authenticated) users determines who is allowed to request which operation. For example, the therapy parameters or device setting should not be modified without a doctor. A big problem with this authorization are emergency situations, for example a heart attack or any sudden health problem in general. Such incidents cannot be disregarded in IMD world. While in the regular case the implant should be running at all times, in special circumstances such as a cardiac surgery with electrocautery devices, it must be turned off, not to affect the patient's condition in a negative way. As can be seen, the device must be able to determine when such a crucial request should be accepted.

Another one of the most important features of any secure system is *confidentiality*. Because of that, it could be taken as a target by the attacker. Any data, either stored on the implant or the reader, should only be read by authorized parties. Although tampering with the implant is not feasible, the programmer should not carry any valuable information in clear text. That way side channel attacks on the hardware components are not attractive to the attacker due to relatively low potential gain. Additionally, the communication should be fully encrypted. It is necessary to prevent eavesdroppers from obtaining sensitives information. While the process of memory encryption can be reused from different fields, the message encryption is more difficult. A typical approach is based on a symmetric encryption. In addition, a key agreement protocol is used to ensure that both parties have access to the same key. Most common ways to achieve it are based on public key encryption. However, due to limited computational capabilities of the implant, the approach is rarely used. Next goal that could be targeted during an attack is the *integrity*. Any data, stored or transmitted using any communication channel, should only be modified by authorized parties. If the initiator of changes or data source is not validated, it could result in serious security consequences. The messages could be altered, which could cause the IMD to act in a harmful way for its bearer. What is more, the operation software could be changed by anybody with sufficient equipment and skill level, resulting in a similar aftermath. Another very important aspect, especially for the IMDs, that could be targeted is the *availability*. The functionality provided by the implant should be accessible to all authorized parties at all times. The most important in this case is the patient himself, which is in need of the medical functions of the device. Knowing the possible damage, it is an important vector of attack to cover. There are multiple ways to execute a denial of service attack (which target availability) on the IMD. It could become unresponsive after a blockage of the communication channel. One of the most dangerous attacks is the battery depletion attack [32]. It can be achieved by flooding the implant with network traffic, or forcing the device to perform energy expensive operations. Some of the requests can be blocked by the means of authentication and authorization. However, it should be pointed out that mass requests for identity validation could deplete large amounts of power reserve.

A final goal that could be achieved during an attack is the violation of patient's *privacy*. Without proper security mechanisms, simple eavesdropping is sufficient to compromise the patient. It is clear that the information stored and transmitted by the implant is very sensitive. Some of the existing privacy goals include [57]:

- Device-type privacy - It should not be possible to tell what kind of device is used by the patient.
- Medical data privacy - Adversaries should not be able to access the data collected by the device, that is used by the doctors.
- Tracking - Unauthorized entities should not be able to track or locate the patient.
- Device-existence privacy - Unauthorized parties should not be able to determine if a person uses an IMD.

If the adversary is able to compromise at least one of the mentioned goals, they could use the information against the victim. Knowing that a person is carrying an IMD is seemingly harmless. However, it strongly suggests that the patient has some health issues. Being able to tell what kind of device it is, it makes it even easier to determine a part of patient's medical history. In case of tracking, adversaries could use the physical layer, for example by matching a radio fingerprint, to follow the location of the patient. It could be done by using large amount of programmers, that are spread in a way that the IMD stays within the communication range. While tracking could be beneficial in some particular cases, such as patients with dementia or Alzheimer, it poses a serious privacy violation and should be carefully controlled [2].

One could consider the mentioned attack possibilities as unrealistic and not likely to be exploited. However, the threat was serious enough that a politician from the United Stated of America had the wireless connectivity disabled in his pacemaker [66]. If multiple important personalities had their devices modified for security reasons, the public could become too afraid to agree for the treatment involving medical implants. This would cause multiple negative effects on various parties. The manufacturers would notice a loss in the income, resulting in less funds being put into improvements of the devices. But despite the financial loses, the most important consequence would be related to the health of the patients. Doctors would have to find an alternative treatment for some particular diseases, usually less efficient then the usage of the implants. It is clear that studies should be taken in order to guarantee the security of the IMDs, since they are an important medical tool.

## 1.2. Research goals and contributions

Due to the danger of adversaries violating the privacy or health of the patients, there is a clear need for some security mechanisms. In the beginning, manufacturers often kept the design details secret, in order to provide security [44]. Such practice is often called *security through obscurity*, a principle rejected over 100 years ago. A system relying on this model may contain major security flaws, which designers decided not to address. Halperin et al. [31] in 2008 examined the communication protocol between an implant and a reader. They have not found any security or authentication mechanisms, which enabled multiple radio-based replay attacks. Clearly, some changes in the manufacturing of the devices were necessary. To be able to prevent attacks discussed above, manufacturers and researchers started working on proper security measures, while keeping in mind the IMD constraints. The implant life span is determined by the energy stored in the battery. Therefore, there must exist a strong balance between security and efficiency. Any cryptographic primitive should be as efficient as possible, while not compromising reliability and performance of the device. Additionally, the implant must be accessible in case of any unexpected emergency. It creates another tradeoff between security and accessibility. A simple solution would involve the patient to memorize a password, that grants access to the implant. However, if in case of emergency the patient is unconscious, the idea fails. When shifting from memorized passwords to wearable tokens, a problem of property theft or loss arises. A different approach could rely on tattooed passwords. But according to Denning et al. [22], patients have objections against any passwords tattooed on their bodies (including tattoos visible only in certain conditions, for example under ultraviolet light) based on social, religious or personal aspects.

The most common approaches used by the researchers can be divided into 4 groups:

1. **Protocols based on physiological signals.**

   The first group involves the processing of physiological signals. Those time-variant biometrics can be measured separately by the implant and the reader. It can provide the base for identity authentica-

tion [56] or encryption key establishment [68]. Usually, they are based on the *touch to access* policy. It says that the signal cannot be measured from the distance. Instead, the external device must be in contact with the patient. The most noticeable benefits of this type is the fact that no key must be shared between the devices, as well as the fact that the freshness of the key is guaranteed, since each one is based on separate reading. It does not come without any drawbacks. At first, the two readings are not going to be the same. Because of that, the solution involves usage of techniques that limit the amount of false authentications and false rejections. What is more, this signal unifying mechanisms usually consume a lot of energy, making it vulnerable to energy depletion attack.

2. **Protocols based on key management.**

   Another group of of solutions is based on the key management, similar to different environments, like the Internet. Such scheme guarantees high reliability, as it is difficult to break the cryptographic primitives, implemented in the correct way. However, the key management is difficult in case of IMDs. Protocols based on public key cryptography are too demanding in terms of computing and power consumption for the implant to handle, therefore symmetric encryption is preferred. Yet symmetric keys can be easily stolen from the reader, rendering the scheme useless [70].

3. **Protocols based on distance bounding.**

   The main idea behind this principle relies on the fact that access is authorized by the patient being aware of the presence of the reader. Similarly to the first group, they also rely on the *touch to access* policy. Rasmussen et al. [55] proposed a scheme, where reader and implant use two channels for communication: classic wireless radio-based, and ultrasound based. Ultrasounds are used to be able to determine the delay between messages, which is used to calculate the distance. However, this scheme can leak some information for attackers with sensitive antennas [43], as well as the fact that it requires the manufacturers to add additional modules for the constrained implant.

4. **Protocols based on usage of external tools.**

   The last group of the main ideas for IMD security is usage of external devices, carried by the patient. They may act as a signal jammer that prevents any communication to the implant [21]. Opposite to this idea, an external device may act as an access station for the implant. One example introduces a smart-phone, carried by the patient, that performs cryptographic tasks, to reduce the energy usage on the IMD. However, as mentioned above, this model fails when the external device is lost - which can easily happen in case of an accident.

As shown, there are various ways of securing the IMDs with different approaches. Each approach comes with different set of assumptions, which sometimes are not easy to implement in real life scenario. Most of them assume that the reader is used only by the authorized personnel. This assumption is not easy to fulfill, as the reader, being a relatively small device, can be stolen. Therefore an adversary could obtain any key material stored inside it, or use the reader to reprogram an implant of an unaware patient. Aware of this problem, in this thesis we try to answer the following research questions:

1. **What is the effectiveness of existing security solutions for the Implantable Medical Devices?**

   (a) How many different attack vectors do they cover?

   (b) Are there any major vulnerabilities in existing solutions?

2. **Is it possible to create a novel protocol, that outperforms existing solutions in terms of security level and resource usage?**

   (a) What are suitable cryptographic primitives for IMDs?

   (b) How can we cover additional attack vectors, such as theft of the reader?

The primary goal of the presented work is to provide a novel authentication and key agreement model, that is easy to use, viable for real life implementation and provides high level security. The main scientific contributions include:

- *Comparison of existing protocols* - we provide comprehensive study of the state of the art solutions in the field of security protocols for the IMDs. We perform comparison of different principles behind that diverse solutions, depicting the main advantages and drawbacks. Additionally we provide a security analysis of chosen protocols to showcase possible risks that included in the design of such systems.

- *Implementation of a new attack on the state of the art protocol* - within the mentioned security analysis we have discovered a possible vector of an attack against a protocol based on physiological signals. We have proved that the signals in time do not vary to an extend granting sufficient security level. We performed an experiment on real patients' data, which showed that the presented work can be broken with effectiveness reaching 70%.

- *Proposition and evaluation of a new security scheme* - after showcasing multiple flaws in existing work we have introduced a novel scheme that mitigates the existing issues. The proposed protocol based on lightweight cryptographic operations is designed to have high security levels, low energy consumption and low implementation costs. Finally, we have performed a set of simulations to provide the most efficient cryptographic primitives and estimate the total resource usage.

## 1.3. Thesis layout

This thesis is organized as follows. In Section 2 we present the related work in the IMD related security. In the beginning we discuss techniques used in digital authentication and key establishment. Later, we show possible solutions other researchers proposed to enhance security and privacy of the communication protocols. We divide them, based on the main design principle that was used. In Section 3 we perform evaluation of some chosen protocols, to show what benefits and drawbacks they bring. In addition, we discuss the implemented attack on the Physiological signal based key agreement protocol, introduced by Venkatasubramanian et al. [68]. Section 4 is dedicated to the new scheme we have created. We depict the necessary building blocks, threat model and the authentication protocol. In Section 5 we evaluate the proposed design. We provide detailed instructions for implementation of the scheme, including energy usage comparison between different cryptographic primitives. Finally, we conclude the thesis in Section 6.

# 2

# Background and related work

The process of electronic authentication and key management is a difficult one in the digital world. In the beginning of the chapter, we provide an overview of basic principles how it can be achieved. Later, we will demonstrate how the application of those principles works in the world of IMDs. In general, the given environment is more difficult to work with, comparing to other ones, eg. the Internet. It is the case due to numerous constraints such as restricted computational capabilities and limited battery capacity. Finally, we present in detail how physiological signals find different applications in the implant protection.

## 2.1. Background

Authentication and key establishment protocols are a foundation for security of communications. In the following section we provide an overview of the main principles for the designers of such schemes. Additionally, we describe chosen exemplary protocols to further elaborate on important aspects and possible techniques.

### 2.1.1. Digital authentication

Authentication is one of the essential building blocks for securing electronic communication. Cryptographic algorithms that guarantee encryption and integrity do not provide any help, if wrong people are in possession of secret keys. The general setting for an identification protocol involves a prover and a verifier. The verifier is presented with the purported identity of the prover. The goal is to confirm that the prover is in possession of the secret key, which should be distributed only to selected parties. Menezes et al. [36] propose following formalization of the authentication:

> *Entity authentication technique assures one party (through acquisition of corroborative evidence) of both the identity of a second party involved, and that the second was active at the time the evidence was created or acquired.*

From this definition, it is relatively easy to deduce two main goals of the process. Firstly, the verifier of an entity is convinced about identity of the other party, accepting the authentic prover. Secondly, he is sure that the proof was not prepared nor used before, but was freshly generated. Authentication scheme, to be considered effective, must fulfill more objectives. No one, including the verifier, should be able to reuse a

successful identification, in order to impersonate the prover from recorded run of the protocol. What is more, no third party distinct from the prover, should be able to impersonate him. Entity authentication techniques may be divided into multiple categories, depending on what feature the protocol is based on. The three best known characteristics include:

- Something you know - Prover and verifier share a secret, that validates prover's identity. Examples include passwords or private keys.
- Something you own - Prover is in possession of a physical item, used for identification. Examples involve passports, smart cards or electronic tokens.
- Something you are - Property in general related to human individuals. It includes physical characteristics and involuntary actions (biometrics), such as fingerprints, retinal patterns, heart rate or handwritten signatures.

The most commonly used pattern to authenticate a party is called challenge-response protocol. The idea behind is that the prover demonstrates his identity to the verifier by showing knowledge of a secret that is associated with his entity. Usually it is done without revealing the secret itself during the message exchange, to prevent eavesdroppers from stealing the identity. A common solution is based on providing an adequate response to a challenge, that changes in time.

### 2.1.2. Key establishment

Another essential tool in establishing secure communication between multiple entities is key establishment. In the early literature on cryptographic protocols it was common to confuse authentication protocols with key establishment protocols. Every design describing a way of setting up a session key between two or more parties was called authentication protocol. The main difference is that some key establishment does not necessarily provide entity verification, while authentication can be done without an involvement of session keys. Menezes et al. [36] provide following definition for it:

> *Key establishment is a process or protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic use.*

The main goals of the key establishment protocol are:

- Key freshness - involved parties know that the key was freshly generated, therefore it is highly unlikely to be known beforehand by the adversary.
- Key quality - key is generated in a proper way, from a source of high entropy, therefore it is difficult to be guessed.
- Key exclusivity - key is known only to the parties participating in the protocol - listening adversary should be unable to recover it.

### 2.1.3. Exemplary protocols

Together, key establishment and authentication build a backbone of secure communication. Some protocols focus on only one of those tasks and must be complemented by additional tools. There are multiple ways the tasks can be achieved. There are two main groups of methods used. The first one is based on symmetric, second on asymmetric encryption. Both come with a set of advantages and drawbacks.

For entity authentication using symmetric algorithms, the prover most commonly performs some kind of cryptographic operation requested by the verifier involving usage of the secret key on data provided by the latter. The verifier subsequently compares the received outcome with the results generated by itself. If those two correspond properly, it indicates that the prover must have the secret key, which belongs specifically to its entity. One of the model examples is showed in standard ISO/IEC 9798-2 [62]. Verifier sends a randomly generated number, while the prover responds with a message obtained by encrypting the received number. The random number ensures the freshness of protocol, prevents the replay attack. In addition, no message reveals any information about the secret key. As with most of the schemes based on symmetric algorithms, the main challenge also in this case is the key management. If the above scheme was commonly used, every distinct pair of users of the system should be in possession of a unique key, which is clearly not feasible. One idea to solve this issue involves usage of an on-line trusted third party. One example of such protocol is Kerberos [49]. The protocol involves three parties: two parties wanting to communicate with each other A and B, that do not possess a common key, and an authentication server. Server owns a secret key shared between itself and every user of the system. At first, party A contacts the server, sending message with its identity, identity of the second party B and a freshly generated random number. Server responds with a message containing the session key and its expiration time. A part of the message is encrypted using the key known by B, to ensure that A was not able to modify it. Finally, A forwards the necessary part to B, so both parties have knowledge of the session key. The Kerberos protocol has key establishment as its goal, in addition to entity authentication. From the point of view of A, the key is properly generated and not leaked due to established trust with the server. To ensure the key quality from the second point of view, the scheme relies on the expiration time contained in the messages. As long as the message has not expired, B can be sure that the key is fresh and well generated. There are some drawbacks to the system. For example, parties involved in the protocol must provide synchronized and secure clocks. If all clocks are not synchronized with the server clock, the authentication could fail due to the expiry date included in the response generated by the server.

The second group of protocols relies on usage of asymmetric encryption. It is in general admitted that there are two main advantages of those protocols over those based on symmetric cryptography. The first is that public key systems allows the straightforward definition of digital signatures. The second one is simplification of key management. It is possible to transfer the data without a trusted third party, which is typical for protocols based on symmetric solutions. While the public keys do not have to be confidential, their integrity must be maintained. The common solution is usage of certificates, signed by trusted authorities. In exchange for the mentioned benefits, asymmetric algorithms bring new costs. The main one is high computational cost, usually two or three orders of magnitude more then symmetric cryptography. While the performance of typical desktop computers have increased to the extent that the delay would not be noticed, low-power computing devices or servers handling multiple clients would find the overhead significant. Public key techniques may be used in challenge-response based identification, where the prover must demonstrate the knowledge of the private key. Most commonly, it is done in two manners: by creating a digital signature of the challenge, or by decrypting a challenge encrypted with the public key. A basic protocol following the firstly mentioned idea is presented in the international standard ISO/IEC 9798-3 [63]. Verifier sends a challenge in form of a random number. Prover replies with a message composed of the certificate proving his identity, a random number generated by them, identity of verifier and a digital signature. The signature is composed over both random numbers and identity of the verifier. In order to verify the claimant, one must check that their own identifier was used, and using a public key of the prover (that can be obtained from the received certificate)

validate the signature. It should be noticed, that a random number generated by the prover is included in the protocol to prevent chosen plaintext attacks.

## 2.2. Authentication in medical devices

The authentication protocols are a broad aspect of digital security. They should be selected properly, based on the circumstances. In the case of IMDs, one of the main concerns is the energy consumption. In this section we provide some of the ideas, based on which researchers try to accomplish the mentioned process.

### 2.2.1. Usage of physiological signals

One of the most common trends in the field is the usage of dynamic biometrics as an authentication mechanism. Rostami et al. [56] made one of the first significant contribution in the area by publishing their protocol called Heart-to-Heart (H2H). The main idea behind it is the measurement and usage of the electromagnetic waveform produced by a heart, called electrocardiogram (ECG). Both implant and the reader would acquire that signal and establish the mutual authentication from it. At first, both IMD and reader would establish a secure channel using lightweight public-key cryptography. This step is similar to TLS protocol, where the IMD would act as a client, and the reader as the server. After the channel is established, both sides read the ECG signal, and send a commitment to the other party. The commitment is bonded to the already established channel, to prevent the re-use of messages. Authors of the discussed paper have noticed that the readings of a signal by two different devices will provide similar, but not identical results. After the readings are revealed, both devices evaluate them using Neyman-Pearson lemma [50]. The procedure is used to evaluate how two different signals correspond with each other. In other words, the differences are small enough that it can be assumed they come from the same source, which is the patient's body, instead of being artificially crafted in an attempted attack. By adjusting the parameters of false negatives and number of inter-pulse intervals used in the protocol, it was possible to achieve reliable system. However, it contains some serious security issues, discussed in Section 3.

Usage of ECG signal as a form of authentication was taken forward by Peter et al. [54]. In their work, authors showed an implementation of the system in the Body Area Network (BAN), that considers timing and data uncertainties present in such an environment. At first, a special sensor board was constructed, that is small enough to be used in BANs, and whose accuracy is suitable for authentication. It consists of a differential amplifier, a filter and a post amplifier. After the signal is acquired, digital signal processing is applied, in order to remove any major anomalies left after the analog filtering performed by the board itself. The peaks in ECG signal are detected using Pan-Tompkins method [52], which is additionally enhanced by a custom validation algorithm. The algorithm is based on exploiting ECG data redundancy and knowledge about typical heart beat. The main idea behind the approach is validation that detected values in the peaks are in the expected order, and if the magnitudes together with time differences are within the expected range for a standardized ECG signal. The authors presented a set of tests, that showed efficiency of the algorithm being close to 100%. After the data is extracted, it is hashed together with an unique nonce, that was randomly generated by the second node. Consecutively, hashes are exchanged, what is followed by exchange of actual data. The final step of the protocol is to verify if the previously received hash was created in an genuine way. In order to reduce number of false negative authentications, authors suggested configuration of eight samples of 8-bit integer precision each. The paper shows an enhanced way of collecting a ECG signal, together with in-depth details of implementation of the system.

ECG can be used in aspects of security different than authentication. Another usage of it is involvement

in key-agreement protocols. One scheme for body area networks was presented by Venkatasubramanian et al. [68]. The main idea behind it is that a cryptographic key is hidden using physiological signal features and a *fuzzy vault* cryptographic primitive [34]. At first sensors collect the physiological signal-based features. It is done by acquiring the samples, transforming them using fast Fourier transform, and finally converting into a format suitable for fuzzy vault. In a result, both sender and receiver possess their own feature vectors. After that, sender node is ready to create the fuzzy vault. In the beginning it chooses a polynomial of publicly known order and random coefficients. The coefficients, concatenated together, form the secret key that the sender wants to communicate to the receiver. The vault is created by applying the polynomial function on the set of features, and adding a much larger set of random points that do not belong to either set of samples. Lastly, legitimate and fake points are permuted and sent. The receiver, in order to recover the secret key must apply the polynomial function on his set of samples, identify common values and reconstruct the polynomial using Lagrange interpolation. The described protocol is unique, as it does not require any pre-configuration steps. Additionally it provides more efficient solution than the Diffie-Hellman key exchange protocol. The protocol does not require any heavy computations, such as the exponentiation operation. The main drawback lays in the size of vault, that must be transmitted. If it is assumed that a system needs 30 feature points, approximately additional 1000 random points should be added, to reduce the chances of attacker in guessing which are legitimate. Knowing that each point consists of two values, the transmission could reach several dozens of kilobytes. In Section 3 we discuss in detail the security of the protocol, and propose a novel attack using patient's medical data.

Another scheme for the key exchange based on ECG was presented by Seepers et al. [60]. Their work shows an alternative approach to the problem that is capable of mitigating the issues described above. Instead of the fuzzy vault authors decided to use *fuzzy commitment* primitive [35]. In the scheme, first both sender and receiver collect the ECG signal. From each time interval, calculated between detected pulses, a predefined set of bits is selected and Gray coded [27], to reduce the disparity between the measurements. Those values are concatenated together, and form one set of bits called the *witness*. While fuzzy commitment is able to deal with inter-sensor variability, the problem could arise that heartbeats are incorrectly detected, which results in shifted pulses. To solve the problem, the authors introduce a classification algorithm, that marks if a misdetection has occurred within the block. Messages are exchanged, to signal that a block must be replaced by a fresh measurement. After the witnesses were generated, the sender applies an error correcting code on the encryption key, and performs an XOR operation between the result and the witness. The acquired value is sent to receiver together with a hash of the secret key to enable verification of decommitment. In order to receive the key, the receiver uses his own witness in an XOR operation and reverts the effect of error-correcting coding. Based on the tests performed by the authors, the protocol's energy usage is suitable for real world implementation. In addition, it greatly reduces the size of messages compared to previous work. The main drawback of the protocol lays in the time of execution. To achieve desirable security and reliability, the process of measurement collection should take about a minute. Since the solution is mainly proposed as an emergency access mode, such delay is problematic for the paramedics at work.

It is also possible to derive cryptographic keys from ECG signals. González-Manzano et al. [29] showed in their work, that it is possible to create a time-invariant key, which is unique and difficult to reproduce. The key is created by any pseudo-random process, that must be provided with a seed consisting of enough entropy, so the key cannot be replicated. The proposed system requires a user wearing any device capable of measuring the ECG signal, and to carry an additional hub, such as a dedicated transmitter/receiver, or a smartphone. In the setup phase, a model ECG of the patient must be created. If a user wants to encrypt data, a set of ECG

samples is collected. First of all they are cleaned of noises, such as those caused by respiration and power line. The following step is feature extraction. For that task, Walsh-Hadamard transform [8] was chosen, because of high computational efficiency. The seed is computed by performing similarity analysis of the model ECG of the patient, and set of samples. It is driven by two parameters, called *discard threshold* and *tolerance margin*. The first one is used to discard features with significantly small values, therefore too unstable to be involved in the process. The latter parameter indicates the tolerance margin to consider that a given feature from the model and the sample are close enough. As a result, authors have achieved time-invariant keys that can be used to encrypt any data belonging to the user. By adjusting the mentioned parameters, the algorithm was tuned to provide high entropy. Key generation is relatively fast (total process takes less than 100 ms) and space efficient, what was shown by performed tests. The proposed work does not achieve perfect uniqueness, but a very good one - results showed that over 95% of users are able to produce unique keys.

While the work described above was a good solution for data encryption, preferably for longer amount of time, it was not suitable for creation of a secure channel used in communication. To be able to achieve it, one needs a system that is capable of generating fresh session keys, that differ from each other, while the work described above excels at generating time invariant keys. Such solution was proposed by Choi et al. [15]. In their paper, they describe a key agreement scheme for Body Area Networks based on physiological signals. The major uniqueness of proposed protocol lays in its efficiency. Compared to previously introduced protocols, authors claim they have achieved savings of up to 90% of power consumption due to reduced message size. At first, body nodes exchange a seed value, that will be used to guarantee key freshness. Secondly, sensors extract the physiological signal. As discussed before, those measurements are very unlikely to be identical. Therefore, the regression analysis is performed by nodes. It involves appliance of the discrete Fourier transform and an algorithm for local maxima detection. The last stage is application of the function on the seed values, exchanged in the very first step of the protocol. Usage of regression function, instead of instance fuzzy vault scheme, allowed to reduce message size to a range of a hundred bytes. The main limitation of the work lays in the reliability and necessary runtime of the protocol. Due to usage of statistical technique, false negative rate is larger then in other schemes. In order to mitigate the issue of nodes failing to agree on the session key, time of physiological signal collection should be extended, up to almost half a minute. While for BAN this is not a significant problem, implementation of the scheme in IMDs could be troublesome.

In this section we have shown how the physiological signals can be used in the security of IMDs. In Table 2.1 we demonstrate the summary of the related work that was presented. All of the described protocols make use of simple access-control policy, called *touch to access*. They assume that the biometric data cannot be collected without attaching physical measuring devices to the patient. However, more and more researchers present different ideas of measuring heartbeats without touch. While the main goal of the research is to improve the health care, it may also affect security of the medical devices. Haan and Jeanne [20] proposed a way to read the heart rate using a simple camera using remote photoplethysmography (rPPG). Photoplethysmography is an optical technique to monitor various vital signs, for example pulse rate. It relies on detecting optical absorption variations of human skin due to variations in blood volume during cardiac cycle. Authors, by analyzing captured video image of the skin in terms of color difference, were able to achieve similarity of over 90% with a contact PPG sensor. Despite that, research performed by Seepers on effectiveness of rPPG shows that "state-of-the-art rPPG methods cannot provide an adversary with an advantage over merely guessing the value of a features generated from the physiological signal, provided that electrical cardiac signals are enforced"[59]. Nevertheless, with increasing interest in the field, the available technology could make all security solutions based on physiological signals vulnerable. Further analysis of some of the protocols is pre-

| Author | Application | Main advantages | Main drawbacks |
|---|---|---|---|
| Rostami et al. | Authentication | Creative comparison mechanism of two distinct signals | Found to be insecure |
| Peter et al. | Authentication | High reliability | Requirement to use a custom hardware |
| Venkatasubramanian et al. | Key agreement | Low computational cost | Transmission of large amounts of data |
| Seepers et al. | Key agreement | Greatly reduced message size | Long time of execution |
| González-Manzano et al. | Key derivation | Ability to generate time invariant keys from a physiological signal | Not suitable for derivation of multiple keys |
| Choi et al. | Key derivation | Energy preservation due to reduced message size | Lack of reliability |

Table 2.1: Physiological signal based protocols

sented in Section 3.

### 2.2.2. Protocols based on key management

A common way of solving authentication related problems is a proper key management. We are able to split the proposed solutions into two groups: protocols based on symmetric and asymmetric encryption. Hosseini-Khayat [33] has proposed a low power authentication protocol, based on block ciphers. Both the implant and the reader are assigned a secret key $K$ during manufacturing process. Additionally, the key is handed to a doctor responsible for the patient. During the communication, messages are created with a requested query, the implant ID and a counter value. The usage of counter is an alternative solution to a nonce, that prevents the replay attack. It should be pointed that the components which form the message are combined together, so the state of the counter is not visible without decryption of the message. Since the block ciphers operate on a fixed-size data, it is crucial to ensure that all block contain parts of the ID and counter value. Without that, it would be possible to swap a legitimate query for a malicious one. In order to do ensure legitimate query, the ID and counter value are concatenated together, and the obtained bit array is interleaved with the bits of the query. This procedure ensures, that after splitting the message into blocks of required size, each of them contains the values that ensure the freshness of the session. Authors propose usage of lightweight block ciphers, such as PRESENT [12] or KATAN [19]. That way the protocol requires as few hardware modules as possible: circuit with the cipher logic, 32-bit counter and 32-bit register holding the ID. While the scheme is secure to passive adversaries and very energy efficient, it does not come without some issues. At first, the implant is bounded to a single reader. While the assumption is feasible for private home station or one located in the hospital, it does not provide a way to access the implant in case of an emergency. Even after providing a valid secret key to an external reader, it would have to guess the correct counter value. Secondly, the reader is vulnerable to physical side-channel key extraction attacks. Adversary that is able to read the memory content of the reader gains full access to the device. Recovery from such situation is almost impossible, as it is not feasible to update the secret key in the implant. Most likely, it would have to result in a

surgery and replacement of the device.

Due to the limited power of IMDs, authentication in medical devices is more difficult to be achieved. Solutions involving asymmetric encryption protocols are not very popular among researchers. One of the few ideas, proposed by Wazid at al. [69], is based on elliptic curve cryptography. In their work, they consider a patient to carry a number of implants, together with a *controller node*, a device that is responsible for communication with any third parties. Security between the IMDs and the controller node is established using shared keys, derived utilizing polynomial-based key distribution protocol proposed by Blundo et al. [11]. Only the authentication of users, such as the patient himself or the doctor, uses the elliptic curve mathematics. The authors propose detailed specification for various operations, such as deployment of the scheme, user registration and login, password update or node addition. To evaluate the proposal, authors perform a theoretical security analysis against some of the known attacks, for example *Controller node impersonation* or *man-in-the-middle*, together with formal security verification using the AVISPA tool [3]. This scheme shows limitations of the asymmetric encryption protocols in the world of IMDs. In the described protocol, only one node must be capable of performing the necessary operations. Therefore, in case of applying the solution to IMDs, the implant itself must have increased computational power, which increases the costs of deployment and energy usage.

### 2.2.3. Protocols based on distance bounding

Another way to accomplish authentication is using a distance bounding protocols. They are similar to schemes based on physiological signals that rely on *touch to access* policy in the sense that the reader should be within a small distance of the implant. There are multiple ways to prove the proximity, such as vibrations, ultrasounds or body electrical connectivity. Rasmussen et al. [55] proposed a scheme based on ultrasonic distance bounding. In addition, it provides a key agreement between the parties, based on the Diffie-Hellman protocol [24]. At first, the reader selects the random exponent $p$ and computes the public share $g^p$. The implant, that has been notified about start of the communication, picks a random nonce and transmits it to the reader, while recording the time of send. The reader computes a XOR operation between the nonce and his public share, and sends the result to the IMD via ultrasonic channel. After the response has been collected, the implant is able to verify if the prover is within a small distance by calculating the time difference between sending and receiving the message. If the verification was successful, the process is mirrored. In other words, now the implant will be sending its DH public share and proving that is within the necessary distance. The range of access can be reduced for emergencies to tighten security. When both parties collected the public shares, the session key is computed. The authors have also detailed various scenarios where this scheme can be integrated with existing solutions. They could enable covering additional attack vectors, such as Do attacks, or provide additional authentication. One such example of the latter is the integration of this scheme with those based on security credentials (e.g., password and smart card). While they create a separate set of problems when a credential has been lost or forgotten, it is still possible to access the IMD using only basic version of the protocol. The main drawback of the solution is increased manufacturing cost, as both the implants and the readers must be equipped with ultrasonic communication protocol.

Further analysis of the protocol is presented in Section 3.

As mentioned before, another principle that can be used for distance bounding is usage of vibrations. An IMD security scheme based on this proposition was introduced by Kim et al. [38]. The main advantages of this side channel are: high user perceptibility and close proximity requirement. Vibrations attenuate very fast in the body, hence can only be captured within a very close range. Additionally, the patient would be aware

of any attempt to establish connection with the IMD. However, some researchers consider it not suitable for cryptographic purposes due to relatively high bit error rate of 2.7% [58]. The proposed solution is split into two parts - battery drain resistant wake up and key exchange protocol. The goal of the first part is to prevent adversaries from flooding the implant with information using radio channel. In order to enable that module, one must put a vibrating device on patient's body. The IMD will periodically check in low-power mode if any vibrations are detected, and if so, will switch to normal mode, to make sure that someone is trying to establish a connection. After the wake up has been performed, devices start the key exchange protocol. The reader generates a random key and transmits it using vibrations. Aware of the channel drawbacks, authors propose an error correcting procedure. The implant encrypts a predefined message with the received key, and sends it via radio channel. If the reader can decrypt the message using the generated key, the encrypted communication can start. If that is not the case, the implant chooses a set of ambiguous bit and sends their positions in clear text via radio channel. The reader performs an exhaustive enumeration of all possible values for the received bit positions, and obtains a set of all possible keys. Then, they are tried against the received cipher text. The correct key is used for establishing a secure channel, since the parties must agree upon any random key, not restricted to one originally generated. If all of the keys are not suitable, the whole procedure is restarted with a fresh key. In the experimental implementation, authors measured that the external device must be on the body of the patient, within 10 cm of the IMD, in order to capture the vibrations correctly. Alternatively, the vibrations could be captured remotely using audio channel. Mindful of that possibility, authors suggest usage of a masking sound. It should be band-limited Gaussian white noise that is restricted to the same frequency range as the acoustic signature of the vibration motor, to maximize its obscuring capabilities.

### 2.2.4. Usage of external devices

The last of the discussed principles in implant security involves usage of exterior tokens. They may vary heavily in the performed tasks, such as external authentication credential, signal jammers and providing a communication interface. Gollakota et al. [28] proposed a scheme with a token performing the two latter duties. The main idea behind the project was to develop a protection mechanism for the IMDs without modifying them. To achieve it, they have developed a token called *shield* that is carried by the patient and acts as a proxy. An authorized programmer that wants to communicate with the IMD exchanges its messages with the shield, which relays them to the IMD and sends back the IMD's responses. Additionally, the shield actively prevents any device other than itself from communicating directly with the implant. It does so by jamming messages sent to and from the IMD. The main difficulty in developing this scheme was to enable the device to jam the IMD's transmissions and prevent others from decoding them, while still being able to decode them itself. The design uses two antennas: a jamming antenna and a receiving antenna. The jamming antenna transmits a random jamming signal. The receiving one is connected to a receiver and a second transmitter. The latter is used to generate and send signals capable of canceling the noise. The scheme was optimized to counter both active and passive adversaries. To prove their work, authors have implemented the scheme and performed a set of tests. With proper adjustments, the bit error rate at a passive eavesdropper is nearly 50% at different locations. In other words, adversary's decoding efforts are no more effective than random guessing. Furthermore, even while jamming, the shield can reliably decode the IMD's packets with a packet loss rate less than 0.2%. Additionally, when the shield is present and active, an adversary using legitimate IMD readers cannot extract a response from the protected IMD, even from distances as small as 20 cm. As all token based solutions, the scheme becomes infeasible for the case of an emergency. The patients, whose lives are threatened, are unlikely to take care of the necessary device that secures their implants.

Another scheme that uses external device was proposed by Chi et al. [14]. What distinguishes the idea from others is that the patient uses a common smartphone rather than a dedicated device to perform an authentication with the requesting reader. The scheme is based on a novel approach to the encryption, that is based on compressive sensing [25]. It combines traditional sensing and compressing one into a single process by exploiting data sparsity. The protocol itself is split into different steps. Initially, the IMD creates a secure channel with the patient's smartphone. The patient enters manually the provided master key for current device, which is transformed into session key by using a predefined key generation function. Whenever doctor wants to establish connection with the IMD, they must prove their identity. Doctor sends the public key and the certificate, which are validated by the smartphone. If the process succeeds, the patient's smartphone shares the session key with the implant and the reader. The main clear advantage of the presented scheme is shifting the computational tasks from the IMD to the smartphone, in order to reduce energy usage. Additionally, introducing necessary functionalities to the smartphone eliminates a necessity of patient carrying additional devices. On the other hand, there are notable issues with the proposed idea. At first, the patient must use the phone himself, which might be a problem for the elderly. What is more, it creates a further problem during an emergency. Authors suggest that the paramedics use the patient's phone, which might create privacy issues, or even be impossible due to device's own security systems.

| Author | Main principle | Main advantages | Main drawbacks |
|---|---|---|---|
| Hosseini-Khayat | Block ciphers | High energy efficiency | One-to-one bond between a reader and an implant |
| Wazid at al. | Ellyptic curve cryptography | Formally proven to be secure against multiple known attacks | Large computational costs |
| Rasmussen et al. | Distance bounding | Proven not to have vulnerabilities | Necessity to include components for ultrasonic communication |
| Kim et al. | Distance bounding | Resistant to battery depletion attack | Low reliability due to vibration channel drawbacks |
| Gollakota et al. | External token | Jamming of the incoming signal removes any possibility to connect to the IMD | Inaccessible implant with a lost token |
| Chi et al. | External device | Heavy computational tasks are not performed by the IMD | Not suitable for people not willing to use a smartphone (for example the elderly) |

Table 2.2: IMD authentication protocols

In this section we have presented what are different principles that can be used to achieve the mutual authentication between the reader and the IMD. Each of the principles comes with a set of advantages and drawbacks. The summary of discussed protocols is shown in Table 2.2.

# 3

# Security analysis

When designing a new protocol, there are multiple steps that, if implemented improperly, could create security issues. Some examples include wrong choice of cryptographic primitives, incorrect adjustment of parameters or simply bad order of messages. In order to demonstrate that even state-of-the-art work is not without flaws, in this chapter we present the security analysis of different authentication protocols. We have chosen three different protocols, that rely on distinct principles. That way it is possible to perform a comprehensive analysis of diverse solutions proposed by the researchers. Firstly, we review vulnerabilities in the protocol called Heart to heart (H2H)[56]. We demonstrate that lack of binding between the passwords and the communication session creates large security flaws. Secondly, we analyze a distance bounding based protocol proposed by Rasmussen et al.[55]. We illustrate that proper protection of the hardware is crucial in building a secure system. Finally, we propose a novel attack on the PSKA protocol introduced by Venkatasubramanian et al.[68]. We investigate a particular implementation of a fuzzy vault cryptographic primitive. We test the attack on data gathered from multiple patients, for example with heart arrhythmia disease.

## 3.1. H2H protocol analysis

The steps of the "Heart to heart" (H2H) protocol proposed by Rostami et al. [56] are specified in Figure 3.1. In the beginning, the communicating parties establish a secure channel $S$ using a lightweight public key cryptography. A scheme such as Diffie-Hellman protocol could be used for this task. The authors do not provide many details how they address this problem. It is worth noting, that usage of small exponent could be susceptible to brute force attacks. The rest of the protocol is responsible for the authentication of the reader using touch-to-access policy. First, both parties collect the ECG readings and generate random keys for the commitment. By the means of **Commit**, authors have in mind a commitment scheme, that bonds a message $m$ bound with current channel $S$ under key $\omega$ as stated below:

$$C = Commit((m, S); \omega)$$

After the values are gathered, parties start exchanging the commitments, followed by key that allows the decommitment. Lastly, the signals are compared. When the IMD and the programmer collect their measurements, $\alpha$ and $\beta$ respectively, they are unlikely to be the same. The main focus of the paper is establishing

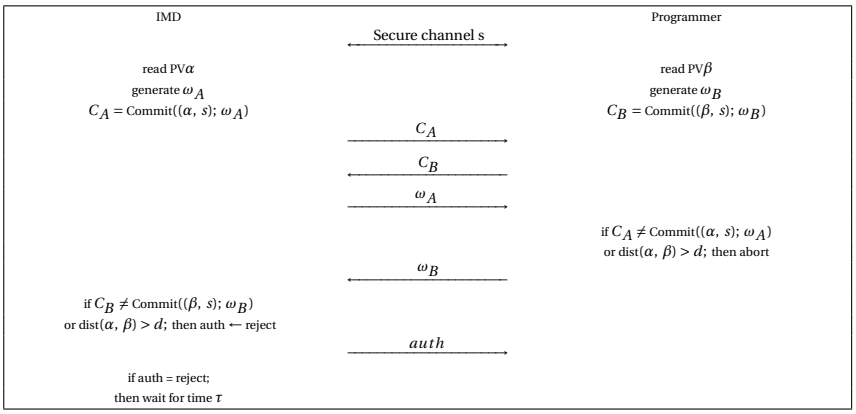



Figure 3.1: H2H protocol [56]

whether $\beta$ is authentic, that is resulting from a measurement made at the same time from the same person, using a device contacting the skin. To verify the claim, the authors use the Neyman-Pearson lemma. This approach is verified by multiple calculations, experiments and mathematical proofs. While this part remains unbroken, it was possible to point serious vulnerabilities in other parts of the protocol.

As pointed by Marin [44], the protocol is fully symmetric. What is more, the implant sends its data first. Therefore it is possible to execute a very simple reflection attack. We demonstrate the attack in Figure 3.2. The actions performed by the IMD remain the same. The attacker, who poses as a legitimate programmer, skips fully the initial steps. Since the threat model assumes attacker cannot make physical contact with the victim, the adversary would be unable to read $\beta$ anyway. Adversary waits until he receives $C_A$, and replays it directly back. The same happens for decommitment step. Because the messages were replayed without any modification, it is clear that they are genuine. The IMD is able to successfully verify the commitment, and establish that the ECG measurement is authentic. Since the proposed protocol is used only for authentication, the parties would start communication in the secure channel established before. There are multiple ways the protocol can be improved to resist the reflection attack. Some of them include:

- Implement a special condition for the implant, to reject the authentication if the signal or the commitment key are the same.
- Addition of a challenge response model, to make the protocol asymmetric.
- Reordering of the messages, to force the programmer to send the values first - this could enable opportunity of attacking the programmer, therefore we suggest combining the step with the previously mentioned ideas.

In addition to the reflection attack, it is also possible to execute an impersonation attack against the H2H protocol. The steps can be seen in Figure 3.3. In the beginning, the adversary establishes two secure channels with the programmer and the IMD separately. Both parties will execute the initial steps simultaneously, assuring successful authentication. At first adversary will follow the protocol with the implant, until they receive both commitment $C_A$ and the decommitment key $\omega_A$. Both values allow the attacker to obtain the measurement of physiological signal $\alpha$. When the adversary is required to send their own commitment, they can send any random data, since the connection is about to be dropped. After acquiring $\alpha$, the adversary will follow the protocol in a regular way with the programmer, by creating new commitment that will bind $\alpha$ to channel $s_2$. Since $\alpha$ is legitimate, the programmer is going to accept the connection, being convinced it is

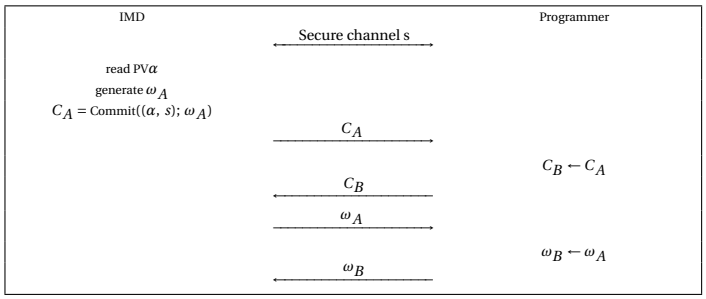



Figure 3.2: The reflection attack on H2H protocol

communicating with the IMD. This form of attack is much more difficult to prevent than the previous one. In a similar situation in the Internet communication, password authenticated key agreement (PAKE) protocols are commonly used [9]. In our case, both $\alpha$ and $\beta$ could be used as passwords. However, as authors of the H2H protocol suggest, there is a major issue with applying this solution in the discussed scenario. The PAKE protocols require exact equality. When dealing with physiological signal it cannot be assured, therefore the parties must check for approximate equality. It is possible to apply one of the fuzzy techniques in order to enable equality testing. This approach was tested by Seepers at al. [59]. What is more, we are able to propose introduction of a timer, to disable a possibility of the programmer waiting longer than a specified threshold for the commitment from the implant. However, such solution can be troublesome to implement, since the delays can be very small and difficult to detect. Also, such threshold could limit the usability of the system because the legitimate connections may be dropped due to connectivity problems.

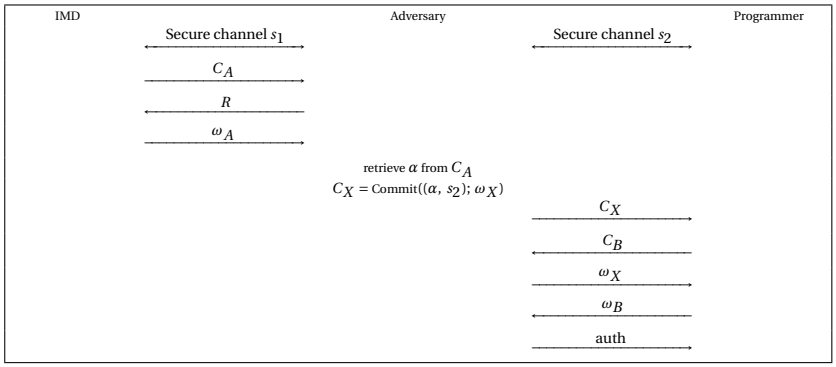



Figure 3.3: The impersonation attack on H2H protocol

The two attacks described by Marin target both parties involved in the model. Therefore it is possible that the adversary can achieve any desirable goal. They can target the IMD to perform data theft or reprogram it to directly affect the patient's health. Secondly, they can target the programmer as well. It poses a threat of providing doctors false feedback regarding the taken actions, which also poses threat to the patient.

## 3.2. Ultrasonic distance bounding protocol analysis

Rasmussen et al. [55] proposed a key agreement and authentication protocol. The key agreement procedure is performed using Diffie-Hellman key exchange. Pure DH does not introduce any form of authentication. On the Internet communication this is commonly solved using certificates. Since this approach is difficult to realize in the world of IMDs, the authors propose ultrasonic distance bounding as a form of authentication.

The outline of the protocol is shown in Figure 3.4.

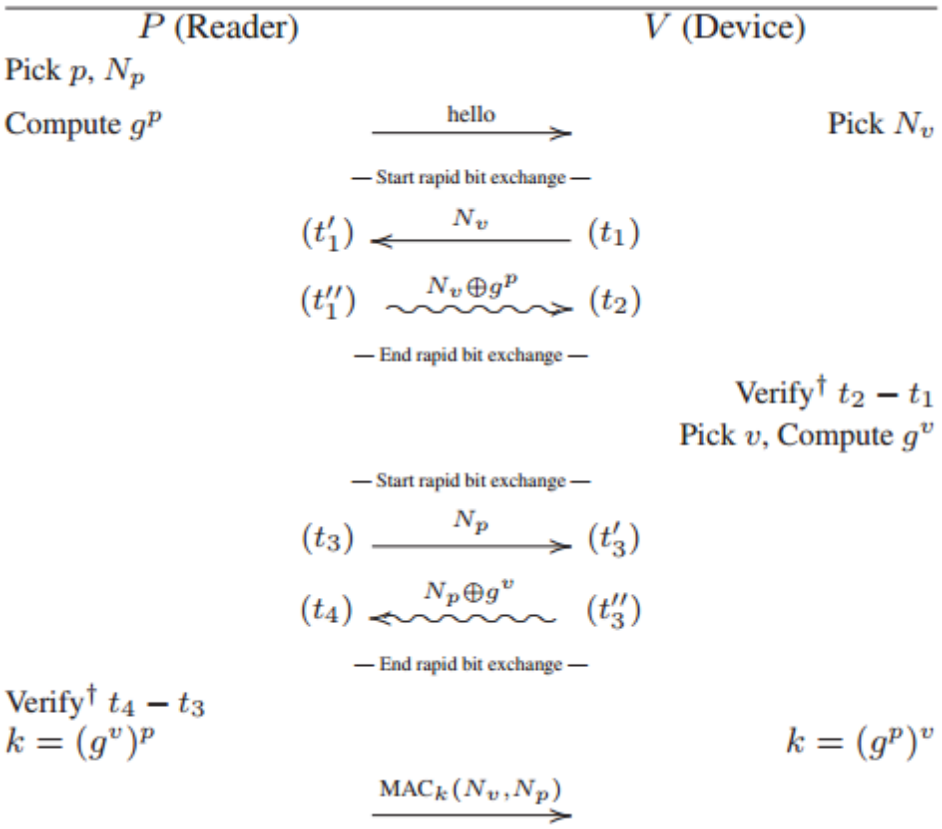



Figure 3.4: Distance pairing protocol [55]

To ensure that a party is within the given range, the DH public shares $g^p$ and $g^v$ are transmitted using an acoustic channel. In order to verify that the parties are in fact within close proximity, the time difference is measured between sending the nonce and receiving the corresponding response. Since the speed of sound is significantly slower than speed of electromagnetic waves, it is easier to avoid the errors in verification. Instead of performing the exchange using two messages, one for the nonce $N_V$ and one for $N_V \oplus g^p$, it is done bit by bit. While the procedure increases the time needed to complete the protocol, it is necessary to prevent distance shortening attacks. One example of attack that is prevented by the practice is a guessing attack [16]. In the discussed scenario, let us assume a *k*-bit nonce that is send as one message. The adversary posing as a reader could form a response by receiving *k-1* bits and guessing the last one. With a 50% probability the attacker guessed correctly and gained a timing advantage equal to the bit period, or in other words time needed for the last bit to arrive. In a similar way, the procedure can be scaled to any number *n* of bits. The attacker waits for *k-n* bits and guesses the last *n* bits with probability of $\frac{1}{2^n}$. The higher the time gain, the lower the probability of executing the attack. The method can be exploited further if the target protocol tolerates a specified threshold of errors. This can be necessary in some situations to increase the reliability.

We can distinguish four main types of attacks on the distance bounding protocols [17]. The attack described above is an example of a *Distance Fraud* attack. In this type of attacks, a dishonest prover will try to shorten the distance measured by the verifier. This type of attack is executed by the dishonest prover alone, without collusion with other (external) parties. Another type of attack is the *Mafia Fraud*, also called a relay attack. In this type of attack, both the prover and verifier are honest, and the attack is performed by an external

attacker. The attacker attempts to shorten the distance measured between the honest prover and the verifier. Usually, it is executed on the schemes that involve usage of private keys used for authentication. Since the analyzed protocol does not involve authentication based on private keys, this attack is not applicable. In the assumed threat model, any party that is within acceptable range of the patient is considered trusted. The next attack class is called *Terrorist Fraud*. In this type of attack, a dishonest prover collaborates with an external attacker to convince the verifier that he is closer than he really is. Since this attack also targets a protocol that uses cryptographic secrets, it is not applicable to the scheme. The last attack is called *Distance Hijacking*. In this type, a dishonest prover exploits one or more honest parties to provide a verifier with false information about the distance between himself and the verifier. Often this type of attack can be carried out by allowing the honest prover to complete the distance bounding protocol as he normally would, and then replacing all messages that contain any form of identity with malicious messages. Since in the analyzed scheme public share of DH key exchange is a part of the distance bounding protocol, the attack is not feasible. The adversary would have to modify $N_V \oplus g^p$ with his own DH public share $g^x$ to form $N_V \oplus g^x$. However, this message would be rejected since the adversary is not within the proper distance.

Acoustic based distance bounding is based on the assumption that the adversary cannot send messages with a speed greater than the speed of sound. While the assumption seems trivial at the first glance, there is a way to violate it. Instead of sending the sound wave, the adversary might create an electromagnetic impulse that will induce an electric current in the receiving circuit. The problem was noted by the authors of the described paper. To mitigate it, they suggest introducing an effective RF shielding. While it might increase the manufacturing costs, it is necessary to prevent the attacker from sending messages at (almost) the speed of light.

## 3.3. PSKA security analysis

In the following section we present a comprehensive security analysis of the PSKA protocol. Firstly, we provide important details about the principles of the scheme. Secondly, we discuss necessity of presence of finite fields in the implementation of the fuzzy vault. What is more, we show disambiguities discovered in the protocol description in the paper. Later, we describe a novel attack against the scheme based on insufficient variance of the physiological signals the protocol is based on. Finally, we provide related work that has been done with regards of the security analysis of the scheme.

### 3.3.1. Protocol introduction

Venkatasubramanian et al. proposed a key agreement protocol based on the fuzzy vault cryptographic primitive. Them main idea behind it, is the application of a polynomial function, whose coefficients form a secret that is sent, on a set of sample features and a much larger set of random points, that do not belong to the set of samples. Person receiving the vault should possess majority of the sample set, to be able to reconstruct the polynomial. This primitive is known to be vulnerable when applied to static biometrics [41]. The authors claim to have overcome the issue by using the physiological signals, which vary in time. Being an example of a regular signal, they can be analyzed in either time or frequency domain. In the discussed protocol, the operations on the signals are performed in the frequency domain. We have decided to test if the difference is sufficient to protect the protocol versus known attacks. The overview of the scheme can be found in Figure 3.5. The communicating parties are collecting a set of features from the physiological signal collected from the patient. Then, the sender transforms the chosen secret into a polynomial, and applies it on a set of obtained features and a set of random points. They are mixed together and sent. The receiver finds the

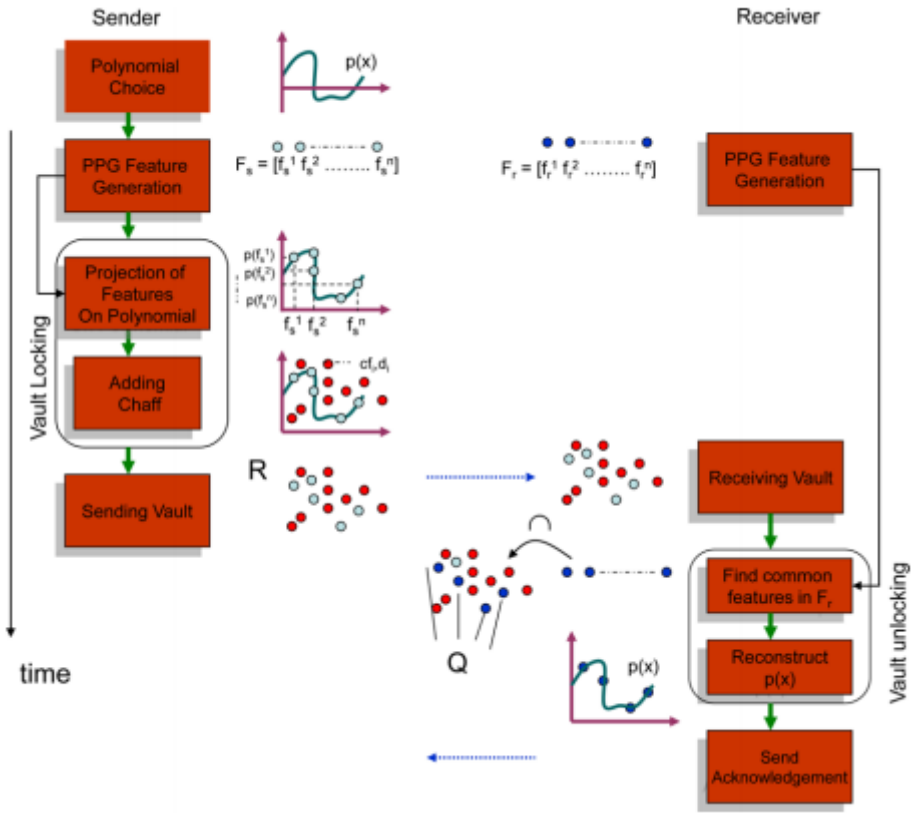



Figure 3.5: PSKA protocol specification [68]

common set of points from his feature generation and the vault it received, and reconstructs the secret using Lagrange polynomial interpolation. In order to accomplish in-depth security analysis we have decided to perform an implementation of the complete protocol. We have tried to follow the instructions from the paper as accurately as possible. However, in some cases the details were insufficient, therefore we had to make some choices, which were evaluated later.

### 3.3.2. Protocol implementation

The first ambiguity we have discovered in the description of the protocol is related to the implementation of the fuzzy vault itself. It is based on applying a polynomial function over a set of features. Later, to open the vault receiver should perform a Lagrange interpolation to retrieve the secret key split into the polynomial coefficients. When we had analyzed the fuzzy vault primitive in detail from the related paper [34], we have discovered that all computations should be performed over a finite field. The authors did not mention this important detail anywhere in the paper. To follow the provided instructions, at first we have implemented the scheme without the finite field calculations. We have discovered two major problems. The first one is related to the floating point arithmetics. The process of unlocking the vault requires creation of so-called Lagrange polynomials using the following formula:

$$P_{n,i}(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

where $n$ is the degree, and $i$ and $j$ define the iteration rules on the set points. The formula introduces a large amount of polynomial divisions, which create errors for the floating point arithmetics. When we have

performed the tests, the introduced errors were recoverable, provided we used 8-bit long features at most. The exemplary results can be seen in Table 3.1. This range is too small for the vault implementation. Let us assume we need to acquire 15 feature points. To properly hide them, we have to introduce additional 500 random points, which cannot possess the same coordinates as the legitimate ones. As the maximum value of an 8-bit number is 256, it can be seen that it is impossible to compute the necessary amount of distinct points, making the scheme non-deployable. When introducing the finite field algebra, the division is replaced by multiplication with the modular inverse of the divisor. That way, it is possible to ensure that the system will return precise values.

| Real key value | Obtained key value with 8-bit features | Obtained key value with 9-bit features |
|---|---|---|
| 32 | 32.0 | 32.0 |
| 13 | 13.0000000009 | 13.0000000149 |
| 42 | 42.0 | 41.999994278 |
| 10 | 10.0000076294 | 9.99975585938 |
| 40 | 39.9995117188 | 39.953125 |
| 78 | 77.9921875 | 75.5 |
| 21 | 21.0625 | 64.0 |
| 46 | 46.0 | 0.0 |

Table 3.1: Sample results of opening fuzzy vault with floating point arithmetics.

The second problem we highlight is related to cryptographic usage of the polynomial reconstruction. This mathematical problem finds usage in multiple protocols, for instance in Shamir's secret sharing [61]. Although the principle works without the introduction of the finite fields arithmetics, it introduces a serious security issue. The adversaries gain a lot of information from each point they are able to find. By knowing the degree of the polynomial, which is a public constant of the system, the attacker can perform the attack from two different sides. The first one involves graphical representation of the polynomial. Based on the known points and the estimated shape of the graph it is possible to decide which points are unlikely to be part of the function. This process reduces the total set of possibilities, enabling the brute force attack. The second approach is related to the mathematical formulas that represent the polynomial. Knowing some values it is possible to derive a set of relations between other coefficient. Since the coefficients are natural numbers, at some points it might be reasonable to limit the possible range of each coefficient, for example when they would reach negative number. Similarly to the previous case, it limits the total number of possibilities. Introduction of finite fields removes both problems. The degree of the polynomial has seemingly low correlation with the graphical representation of the graph. Additionally, finite field arithmetics guarantees that a value will always stay in the range of the field. The rest of the tests were performed with a version that uses finite fields.

| Property | Value |
|---|---|
| Key length | 128 bits |
| Polynomial degree | 7th |
| Finite field | $GF(2^{127}-1)$ |

Table 3.2: PSKA protocol parameters

After we have implemented the finite field arithmetics, the next ambiguity we encountered was the feature creation. The whole process is based on detecting local maxima in the electrocardiogram in the frequency domain. Authors describe that a feature is a binary string, created by concatinating and then quantizing two values: index of the peak and the absolute peak value. The authors have not specified the parameters for neither the peak detection algorithm (how accurate is should be, i.e. how large peaks should be detected) nor the quantization process. After some investigation we have realized that the parameters for the peak detection function are not essential to the protocol, as long as both the reader and the implant follow the exact same algorithm. The situation is different for the quantization process, as inaccurate quantization could result in either no one being able to open the vault or the scheme being insecure. We based our parametrization of this step on a very important information the authors have provided in the text. According to the paper, the windows measured up to 1 second apart should not have changed considerably, thereby still allowing successful unlocking of the vault. Knowing this information we have collected the binary string and performed a set of tests, to see how many most significant bits should be used to fulfill the design choice. In a result, we have discovered that only the four most significant bits of the binary string should be used. As far as the rest of the parameters is concerned, the values were provided by the authors in the experiments they have performed, and are shown in Table 3.2. We used 128-bit keys, seventh degree polynomial and a finite field $GF(2^{127}-1)$.

### 3.3.3. Novel attack description

Since the PSKA protocol is based on the ECG signal, the main idea behind the proposed attack is that the signal does not vary in the frequency domain as much as in the time domain. We supposed that the difference introduced in a small sample is not sufficient to provide satisfactory security while keeping the scheme reliable, i.e. a desynchronization of up to 1 second can happen. As a result, the vault could be unlocked using data recorded in a different time than the authentication attempt. The most obvious representation in a real life situation would be usage of patient's historical ECG data stored in a hospital. Therefore, in our attack model the adversary is in possession of the patient's medical history, for example stolen from a medical facility.

In order to simulate the situation, we have used ECG data from the PhysioBank database[1]. We have collected 10-minute-long signals of 10 different healthy people[2] and 10 of people with hearth arrhythmia[3]. Heart arrhythmia is a heart disease characterized by uneven heart beats. They may be too slow, too fast or in irregular time intervals. Because of that the ECG signal produced by an ill heart differs from a healthy one. The added randomness may provide additional entropy to the signal, making the scheme more resilient. However, on the other hand it may limit the scheme reliability and usability. In the attack implementation the implant is creating a vault using a 4 second longs sample of the signal taken from a random place, and hiding a randomly generated key. The adversary had access to the rest of the signal. They were generating every possible window of the same length. Then, the features were extracted from the sample and used for an attempted vault opening. If the number of recognized features was greater then the minimum required (for the 7th degree polynomial the minimum is 8 features), the polynomial was reconstructed using Lagrange interpolation algorithm. The final step was to check if the obtained polynomial coefficients form the same secret key as the one generated by the implant. If that was the case, the attack had been successful. After initial tests we have found that the attack is possible. In order to learn more about the probability of the attack being successful we have performed a set of tests. The main principle behind them was to measure how likely is

[1] http://www.physionet.org/physiobank

[2] Set: http://physionet.incor.usp.br/physiobank/database/edb/; Samples: 103, 113, 121, 126, 136, 139, 147, 155, 161, 203

[3] Set: https://www.physionet.org/content/mitdb/1.0.0/; Samples: 100, 101, 102, 103, 104, 105, 106, 107, 108, 109

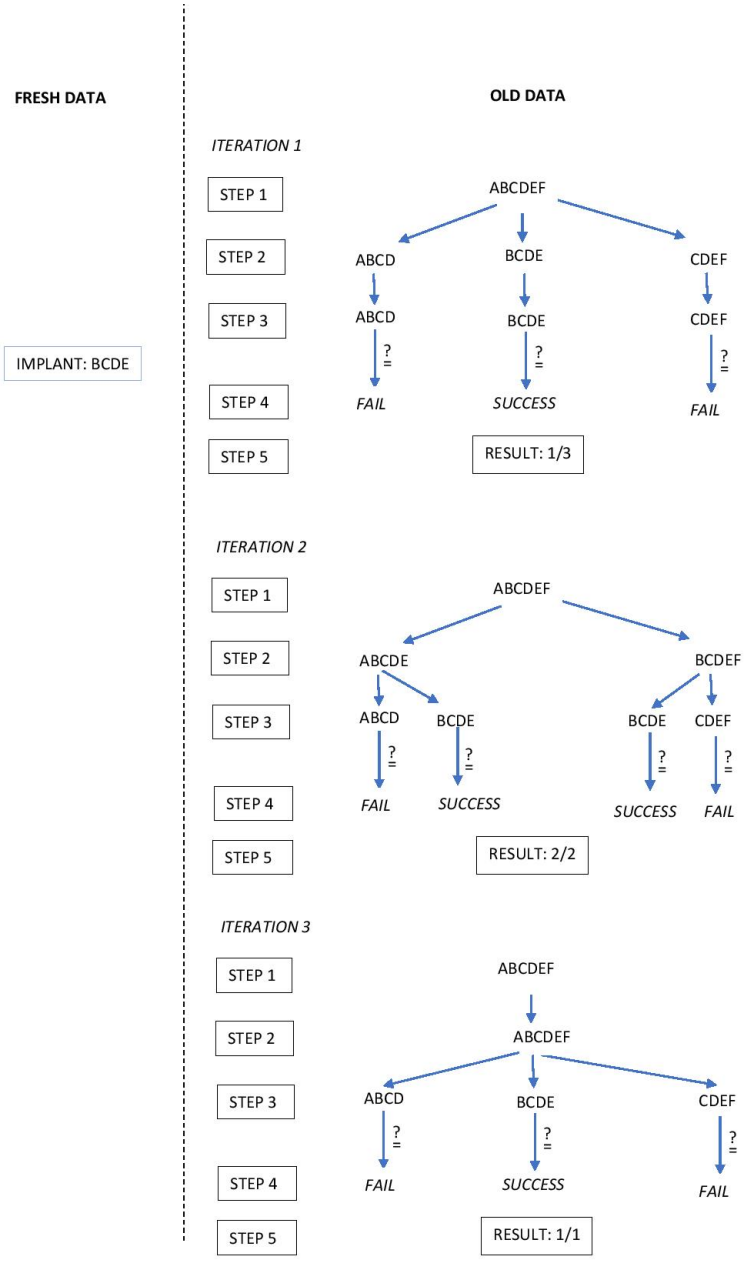



Figure 3.6: Algorithm of the attacker on exemplary data.

the adversary to open the vault depending on the sample size it possesses. The algorithm of the attacker is as follows:

1. Split the possessed signal into every possible sample of minimal length equal to window size.
2. Generate every possible window of 4 seconds in length.
3. Try to open the vault using the sample, and store the result.
4. Calculate how many samples contained a window which was successful in opening the vault, and divide the result by number of samples. 1/3
5. Increase the sample size by a desirable factor and repeat the procedure until the size is equal to length of the whole set.

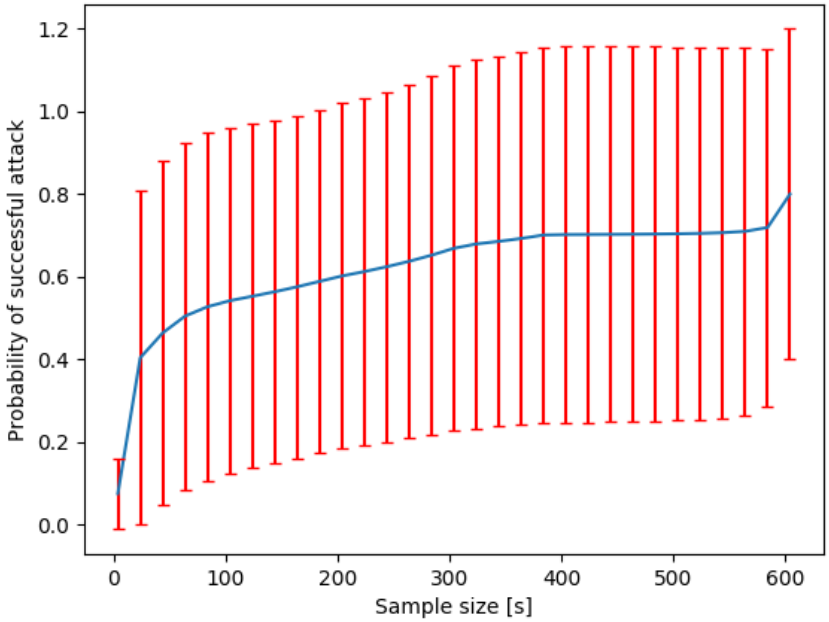



Figure 3.7: Attack probability for 10 patients with a heart arrhythmia.

In order to better illustrate the algorithm we describe it using an example. The diagram of the attack can be seen in Figure 3.6. For simplicity let us assume that a signal of length of $n$ seconds produces $n$ features. During a key establishment attempt the Implant collected a 4 second long ECG sample that was transformed into a set of 4 features *BCDE*. Those features would be later used to create the fuzzy vault with the session key. The attacker managed to steal archival ECG data of the same patient beforehand, for example from a hospital. Let us assume that the stolen data is 6 seconds long and by processing it one is able to acquire a set of 6 features denoted as *ABCDEF*. By checking every possible window of length of 4 seconds, the attacker eventually is able to match the desirable set of features *BCDE* and therefore execute the attack successfully and open the vault. However, in our experiment we try to estimate probability of the attack based on the length of signal data that was stolen. Therefore in the first step we divide the complete signal *ABCDEF* into 4 seconds long samples, as this is the windows size used by the Implant. Then from each sample all possible

windows are created (for the smallest sample size, there is only 1 window per sample). Then each window is compared against the desired sequence. Finally, we calculate the percentage of samples that contained the set of features *BCDE* and increase the sample size. By repeating this procedure until our sample size cannot be increased due to lack of data, we managed to observe that for samples of length 4 we achieved effectiveness of 33%, and for samples of length 5 and 6 to success rate was 100% as each sample contained the wanted sequence. Using the described algorithm we are able to determine how the sample size affects the probability of opening the vault. It is important information, as it determines how much data should be acquired by the adversary in order to successfully break the scheme.

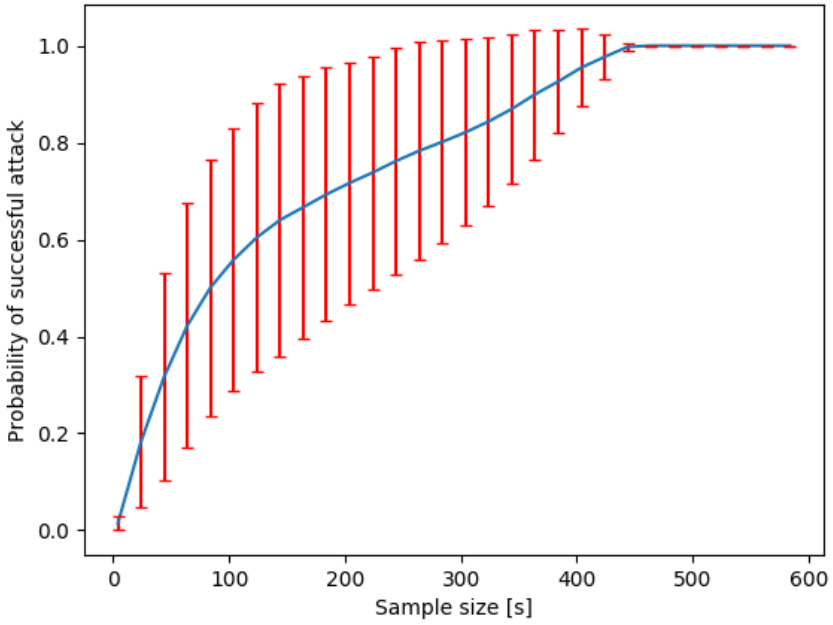



Figure 3.8: Attack probability on a healthy patient

While the behavior of the attacker stayed unchanged, we were modifying the setup by the actions performed by the implant. In the first test, we tried calculating how the probability changes for a single person. To perform this experiment, we have run a simulation where the vault was created by the implant for a single patient[4] 10 times, using sample taken at different non-overlapping places, selected randomly. The results can be seen Figure 3.8. The x axis represents the length of the signal in seconds that was provided to the attacker. The y axis shows the probability of unlocking the vault. The attacker was able to achieve effectiveness of 50% given 85 seconds long samples. We have managed to break the scheme with the probability of 100% given 7.5 minutes long samples. In the next step we have applied the same procedure on a patient with a heart arrhythmia[5]. The results can be seen in Figure 3.9. In comparison to the previous test, here the effectiveness of 50% is achieved for much shorter samples, approximately 25 seconds long. The maximum effectiveness is achieved only slightly faster, at samples of length of 7 minutes. However, a significant difference can be noted when comparing the standard deviations (highlighted in red). We can see that for the patient suffering from a heart arrhythmia the dispersion of the calculated values is larger. In other words, the experiment

[4]Subject 136.
[5]Subject 103.

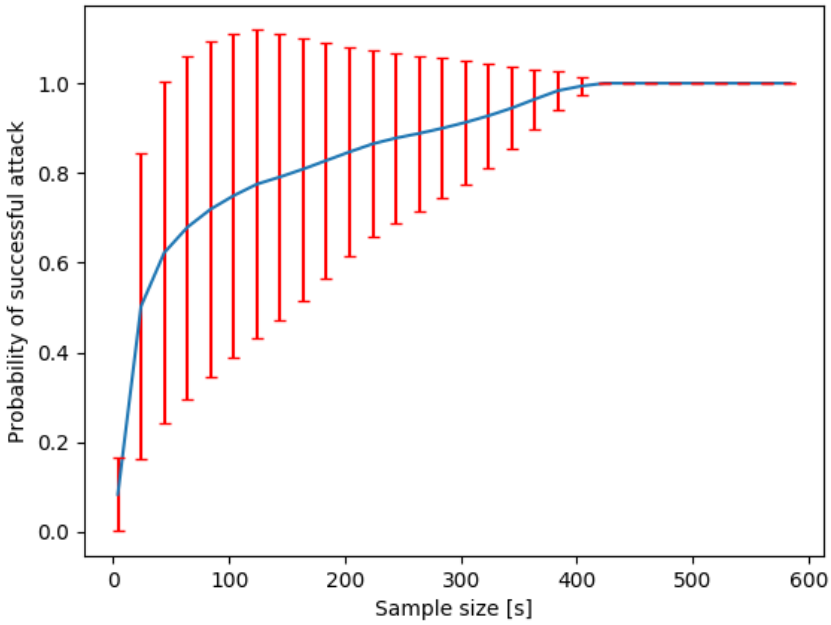



Figure 3.9: Attack probability on a patient with a heart arrhythmia

on the healthy patient generated more uniform results. This behavior could be affected by the additional randomness introduced to the signal by the disease.

When analysing the graphs the first noticeable observation are large standard deviations. There are multiple different explanations for that phenomenon. The first factor is the way our algorithm of calculating the probability works. We are looking for a single instance of positive vault unlocking using a short window taken from a larger sample. Therefore for short samples it might be difficult to find a window that enables successful attack. Contrary to that, for large samples close to maximum sample length the chance of finding a suitable window is larger and more consistent. This is especially visible for the last data point in our graph, where we have only one 10 minutes long sample for each subject. Just a single appearance of a window that unlocks the vault in the whole analyzed signal marks the whole sample as suitable for the attack execution. Additionally to that, grouping of the suitable windows in the signal has significant effects. Since we analyze overlapping samples of data, a single window is present in multiple samples (except for the initial sample length). Multiple suitable windows which were found close to each other in terms of time they were taken, are likely to appear in the same samples. Meanwhile in ranges lacking those, multiple samples will be marked unusable. The situation changes if the distribution of suitable samples is more uniform.

Another factor responsible for large result dispersion is the sampling method we have used. We have taken the samples shifted by at least 1 second. This shift might be too large for our experiment. If the implant records a window at time $t$, a window recorded at time $t-4s$ could be found not suitable. However, a window recorded at time $t-4.5s$ still has a chance to unlock the vault. Let us explain this behavior using an example. In Figure 3.10 we can see an arbitrary ECG signal sample in the time domain. Let us assume the implant has recorded a window starting on the 6 second mark. We can see that the window start is close to the spike called *QRS complex*. If the adversary recorded his sample beforehand at 0 second mark, the windows differ to a large extent. The main difference is different number of the spikes in a window: the one recorded by

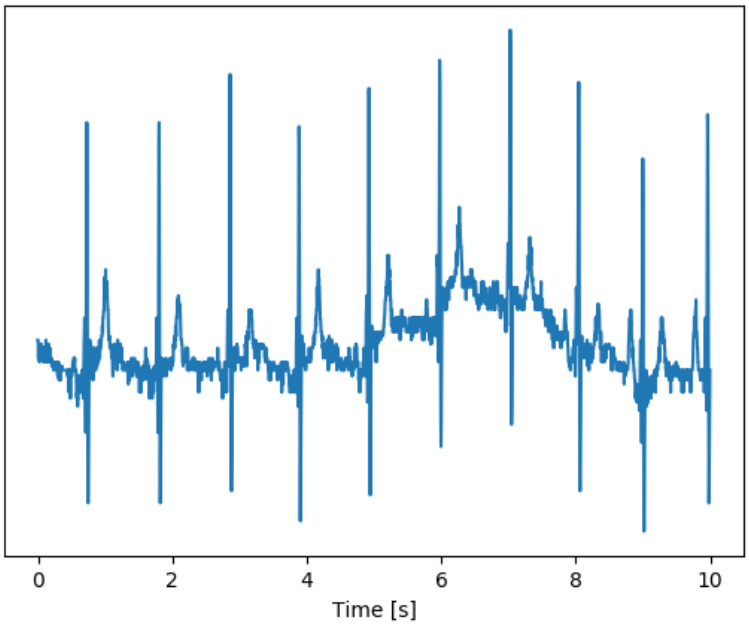



Figure 3.10: Exemplary ECG signal in the time domain.

the implant includes 5 of them, the one captured by the adversary only 4. By changing the starting position by 1 second, the likelihood of capturing similar samples is much lower compared to a smaller shift. This introduces additional randomness to the experiment. When we combine it with the natural unpredictability represented by non deterministic signal produced by the heart, the consistency of the attack is decreased and the standard deviation increased. By increasing the number of samples taken from a single piece of signal by reducing the spacing between the starting positions, it should be possible to achieve better effectiveness of the attack with smaller variation.

The next experiment we have performed aimed to investigate how the probability of a successful attack changes depending on the patient. In this setup the implant was generating the vault from a sample taken at the same random starting position from 10 different healthy subjects. The results of this experiment can be seen in Figure 3.12. The first thing we are able to notice is a high similarity to results generated from a single healthy patient in Figure 3.8. The effectiveness of 50% is achieved with samples of length of 75 seconds, 100% was obtained with 7 minutes long samples. What is more, both the shape of the diagram and the variation in the results is almost identical. Close similarity of the graphs confirms that the presented attack is reliable in its effectiveness. To complement the previous case we have performed the same experiment, but on patients with a hearth arrhythmia. The results can be seen in Figure 3.7. This graph significantly differs from all previous diagrams. While the probability of successful attack of 50% is achieved for similar samples, that is 60 seconds long, the maximum effectiveness reaches only 80%. Additionally, the standard deviation is much larger than for other graphs. It means that the results are not as accurate as for the healthy patients. When comparing the results gathered in this experiment to Figure 3.9, we cannot be confident in the higher effectiveness observed before. For the subject used in previous experiment we managed to achieved very good results, but for other subjects it might be much lower. In order to fully investigate this phenomenon we encourage to repeat the testing we have done on a larger scale, for example using longer samples or by per-

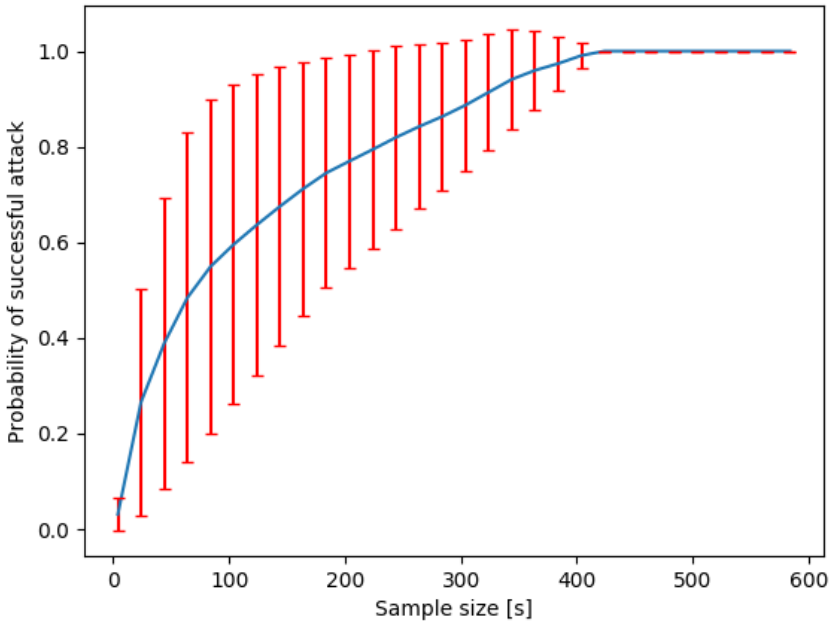



Figure 3.11: Attack probability for 10 healthy patients.

forming the test on larger amount of subjects. Worse reliability of the attack can be justified by large amounts of randomness introduced by the disease. Nevertheless, the attack has very high chance of successful execution.

In another set of tests we have tried to increase the security of scheme. Therefore, we have increased the window size used to create the vault from 4 seconds to 8 seconds with the shift of 1 second. The result it should create a bigger signal difference between the samples, making them more unique, therefore unusable for authentication at a different time than the recording of the sample. We have used the same setup as for experiment shown in Figure 3.8, while changing the window size and calibration of the protocol to ensure that the 1 second difference would not affect reliability of genuine authentications. The results can be seen in Figure 3.13. We can notice that with each window size increase the probability of successful attack was dropping. Compared to the initial size of 4 seconds, we can see that the maximum effectiveness was achieved only for 5 seconds long windows and the dropped down to 60% given 8 seconds long window. Additionally we can see increased values for standard deviation, what means that the consistency of the attack is reduced even more. Based on that experiment we can deduce that the ECG signal in the frequency domain varies more the longer samples we use. What is more, increase of the window size appears as a valid way of improving the security of the discussed scheme. We encourage the readers to extend the experiment in order to find the minimal window length, for which the described attack is not effective anymore.

In this section we have shown the probability of breaking the PSKA protocol using patient's archival data. During our experiments we were able to reach very high attack success rate. For the healthy patients we managed to reach effectiveness of 100% for samples shorter than 8 minutes. Additionally, we have run the tests also on patients with heart arrhythmia. We managed to show that the disease introduces additional randomness to the ECG signal, therefore making the attack less efficient. However, even the lower probability of success managed to reach 80% for 10 minutes long samples. Such high values confirm that the described

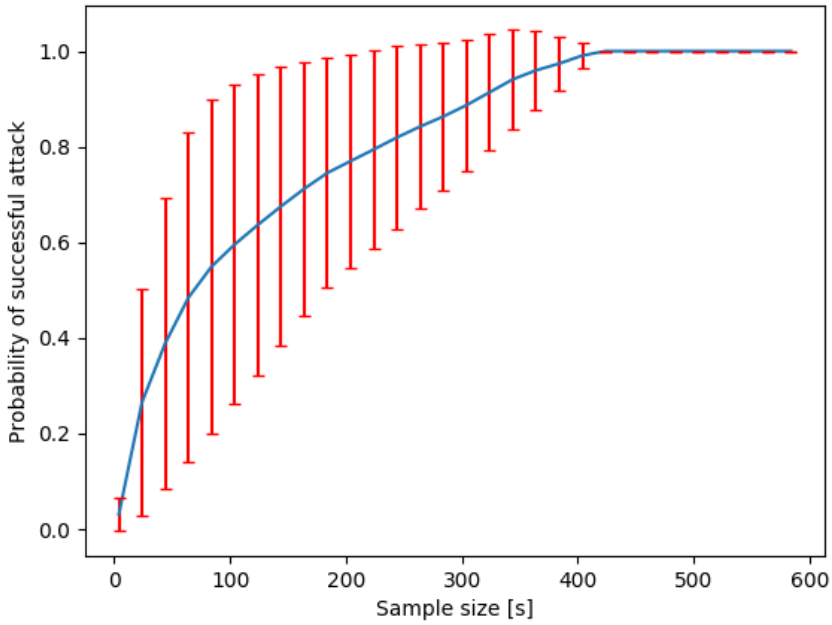
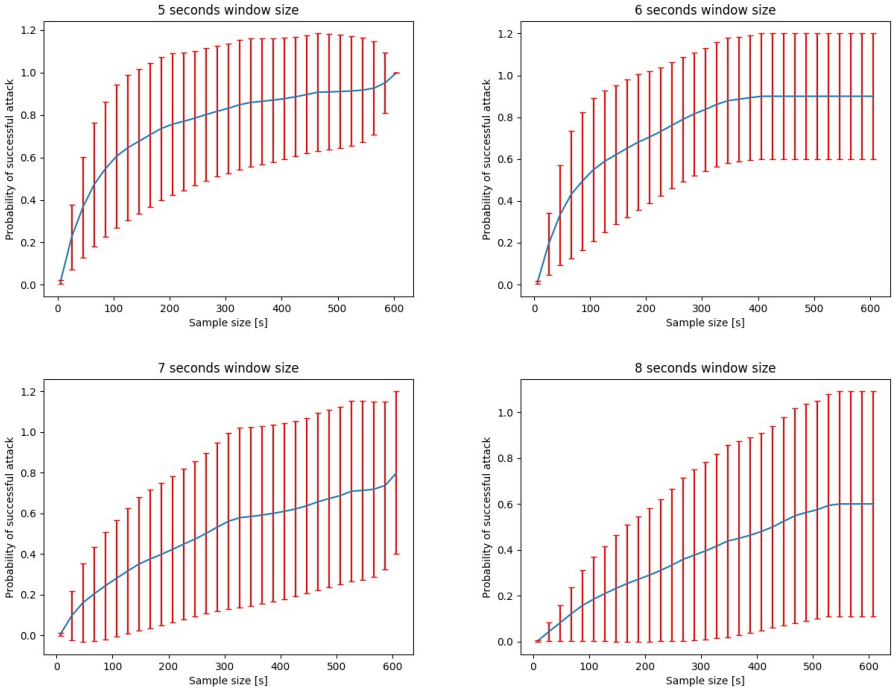



Figure 3.12: Attack probability for 10 healthy patients.

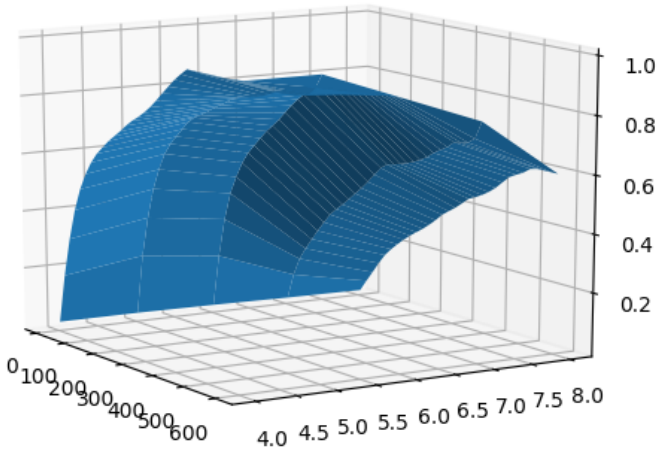
protocol is vulnerable and should not be implemented in its current form to a real life systems.

### 3.3.4. Known vulnerabilities

The PSKA scheme was known before to be vulnerable to other attacks. Bagade et al. [5] managed to perform a model based attack. In this attack, the adversary has access to a generative model of the signal. The attacker generates a signal sample that is diagnostically equivalent to the original signal. The attacker then extracts timely features from the artificial signal and uses the features to open the vault. A generative model is a function of time and physiological parameters of the signal. During the experiments, they were able to perform an attack, given seventh degree polynomial and a 30-second-long samples, with the probability of 30%. At this point it worth noting the difference between the attacks. In the attack we have implemented, the shorter the sample, the easier it is to find a similar one recorded in the past. On the other hand, attack executed by Bagade et al. requires longer samples in order to improve the construction of the model. In contrast to the their attack we were able to achieve better effectiveness, for the protocol parameters as specified by the authors of the PSKA scheme. We have executed the tests on people with and without heart problems. One could argue that our attack is more difficult to execute, since the adversary must be in possession of the medical history of the victim. However, the attack described by Bagade also requires real signals to be able to create the models.




(a) Separated attack probabilities for distinct window sizes.




(b) Attack probability for window sizes between 4 and 8 seconds.

Figure 3.13: Attack probability for different window sizes.

## 3.4. Conclusions

In this chapter we have performed a detailed security analysis of three different protocols. We have decided to select protocols relying on diverse principles. That approach enabled performing a comprehensive analysis of the related work and providing useful guidelines for implementation of our novel solution. During analysis of H2H protocol, we have learned that it is important to apply equal attention to details as to the most important concepts. The authors have put a lot of effort into analysis of the physiological signal, making sure that the number of false positives and false negatives is minimal. However, when talking about security, a single flaw in the system is needed for the adversaries to cause damage. Rasmussen et al. focused on distance bounding using a sound channel. While the analysis have not discovered any major flaws in the proposed solution, one should consider two major problems. The first one is a large energy consumption for the IMD. In general, protocols based on asymmetric cryptography are not popular choice due to large amounts of energy required to compute the exponentiation. The second issue is increased cost of the manufacturing. Both IMDs and readers must be equipped with ultrasound transmitters and receivers, which are properly secured. The PSKA protocol is based on usage of physiological signals, which are handled using a fuzzy vault primitive. We have shown that the idea brings significant energy costs both in computations and transmission. It is caused by the large sample sizes needed to ensure sufficient security levels. Additionally, it was discovered that the scheme if vulnerable due to the nature of the signal that was used. Based on the mentioned findings, we have decided to include the following conclusions in our design:

- *No usage of time invariant physiological signals* - it was shown that the physiological signals must vary in time to prevent adversaries from reusing old patient data
- *No usage of energy inefficient algorithms* - we aim to find a system that does not have to use excessive energy on security related operations, whether it is physiological data handling or public key cryptography (even in the lightweight version)
- *No introduction of specialized hardware components* - we try to find a solution that can be easily introduced in real life situation, where financial costs of the implementation are very important

# 4

# Protocol introduction

In this section we present our novel security scheme for implantable medical devices. We first discuss the existing threat model, that is used in most of the state of the art solutions proposed by the researchers. In order to enhance our work, we have extended the capabilities of the adversaries, creating a more rigorous environment than usual. Next, we discuss the necessary infrastructure and the protocol itself. Because the protocol requires Internet access, at the end of the section we describe a fall back solution. It would keep the system operational, despite some attacks such as distributed denial of service (DDoS) attack on the server. Both the threat model and system infrastructure backed with the knowledge gained in the security analysis help to define requirements for our protocol. We then describe the tools and protocols that were needed to create the scheme. Finally, we propose possible protocol extensions in order to protect the IMD against the batter DoS and enable offline mode operation.

## 4.1. Infrastructure design

In the following section we are going to discuss two important aspects that determine the requirements for the protocol. The first one is the attacker model. It defines what actions are expected from potential adversaries. The second aspect is the system model. It describes the participating devices and actors in the protocol. Finally, we provide a list of requirements for our scheme.

### 4.1.1. Attacker model

In the typical IMD security model the adversaries can be divided into two groups. Passive eavesdroppers can only listen to the ongoing communication between the parties. Assuming message exchange happens via an insecure channel, this type of the adversary may threaten confidentiality and privacy of the patient. By just reading the messages they can determine who is carrying an implant, what is the type of the implant, and finally what is the medical condition of the patient. Another type is the active adversary. They are not limited to capturing messages, but can perform any operation on the communication channel. The most common approach to solving a problem of security in IMDs involves only two parties: the IMD itself and the reader. With that concept, the adversarial model says that both parties are always trusted. It clearly is a reasonable assumption for the implant, since they are produced under strict regulations. Additionally, it is infeasible to

tinker with them after they are deployed. However, the case it not that simple for the reader. Usually, it is a little device. The small size facilitates possible theft and modification of the equipment. In order to solve this issue, we introduce a modified adversarial model. In the proposed design, we assume that the adversary can:

- Capture any message and store it for further usage
- Modify messages on the communication channel
- Forge and insert new messages
- Block messages
- Steal the reader, which enables software/hardware modifications and data theft

By the additional condition, the scheme proposed in this thesis is resistant to new array of possible attacks. However, this comes at a price. The reader cannot contain any information in the plain text. Additionally, we must introduce a way to distinguish two authentication attempts. In the first one, legitimate doctor or paramedic uses the device in everyday job. Secondly, an adversary is using a stolen, but genuine reader to establish the connection. In the non extended version of the threat model, the proposed solutions do not consider the readers to be used by attackers, therefore there is no need to verify the identity of the user. For the sake of simplicity, we do not consider the adversary to be able to block all ongoing communication for a large period of time.

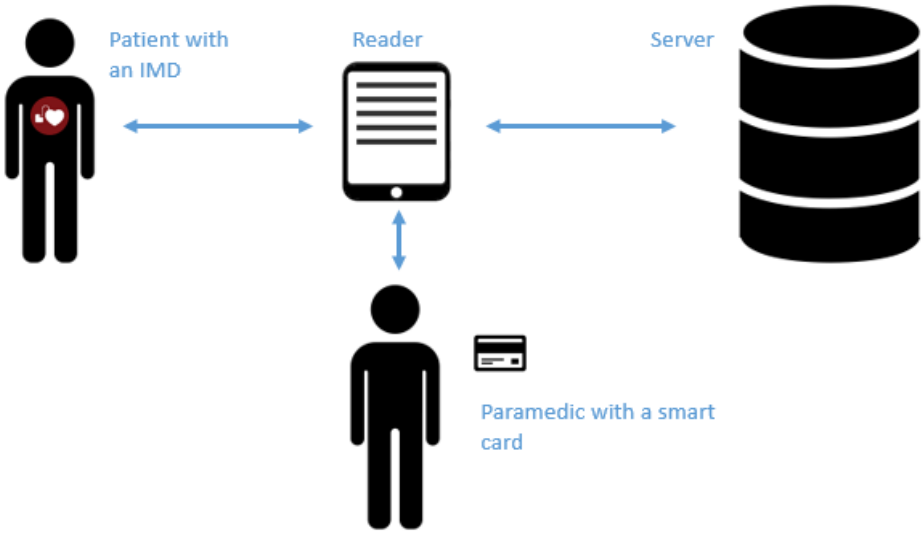



Figure 4.1: General model of the scheme

### 4.1.2. System model

The general model of the proposed solution is shown in Figure 4.1. Apart from the typical presence of the IMD and the reader, we are introducing two new components: a secure server and the identification of protocol users in a form of a smart card. It is responsible for storing data, for example the access attempts to the IMDs. However, the most important functionality is authentication to the implants, which will be covered in detail in Section 4.3. The addition of authentication factor for the users, brings a couple of benefits. Firstly, it is vital for the particular environment. Health care is a sensitive setting, where any changes are monitored

and documented for the good of the patient. In case of possible problems, it is important to know who is responsible for the changes in the treatment. Since most of the IMDs can be reprogrammed to suit individual needs of each patient, it is essential to track people responsible for adjustments. Due to added components in the proposed model, the typical reader must be enhanced by additional functionalities. Firstly, it must be capable of reading a smart card, in order to authenticate the personnel. Secondly, it has to be equipped with a network interface module, that is used to establish Internet connection to the server.

### 4.1.3. Requirements

In order to properly design and later be able to evaluate a security protocol, firstly, we must define a list of requirements the system is supposed to fulfill. Based on the conclusions deducted from security analysis and previously defined threat model and system architecture, the key requirements for the novel protocol are as follows:

- *No sensitive data on the reader* - due to extended threat model compared to related work, we are unable to store any private information on the reader without additional security measures. Any information that cannot be obtained by the adversary, such as encryption keys or patient data, must be either encrypted or not placed on the reader.
- *No sensitive data in the air* - knowing the capabilities of both types of the attackers defined in Section 4.1.1, the communication before establishing a secure session tunnel cannot contain any private information. Even a case of simple eavesdropping poses a large threat if that assumption is violated.
- *Low energy consumption* - knowing the importance of battery life in IMDs it is crucial that every algorithm consumes as little energy as possible. We strive to deliver a solution that uses only lightweight cryptographic operations. However, due to importance of the implants, the reductions in energy usage must not come with a price of lower security standards.
- *Low manufacturing costs* - the aim of the project is to deliver a solution that can be quickly adopted in a real life situation. To achieve this goal the implants should not require specialized hardware that increase their costs. Additionally, our goal is to build as modular solution as possible. That way, it can be easily adapted to different requirements presented by the manufacturers, as well as changing security standards.
- *Presence of an emergency mode* - the IMD should be accessible in both hospital environment during regular treatment, as well as during during emergency situations that can happen in any place. Because of the critical life supporting functionality, there must exist a fallback solution to ensure the safety of the patient. However, it should not introduce new attack vectors to the system, which are easier to exploit.
- *Fulfillment of the core security principles* - the system we propose should conform to the modern security standards. The main concepts involve data confidentiality and integrity, system availability, user authentication and authorization and patient's privacy.

## 4.2. Building blocks

In order to facilitate understanding of the protocol, in the following section we describe the main components contributing to the final solution. We first depict the construction of hash chains, with an example of their practical implementation. Later we discuss the Transport Layer Security protocol. Finally, we reason about

smart card authentication techniques. We explain different available methods, together with their advantages and drawbacks.

### 4.2.1. Hash chains

A hash chain is successive application of cryptographic hash function to some data. It is a common method to derive multiple session keys or one-time passwords from a single piece of information. A popular example of creating one-time passwords using hash chains is a scheme designed by Lamport [42]. In the scheme, hash function is applied repeatedly to a previously chosen random seed. The number of iterations is determined by the desired amount of passwords to be generated (for the sake of this paragraph let us call it N). The final value is stored on the target system, that in the future will be responsible for password verification. A user trying to log into the system has to compute the password, by applying the hash function to the seed N-P times, where P is the counter of logins. The server validates it, by hashing it one more time, and comparing with the value stored. One could notice, that this is a common approach to password based authentication. Storing a hash instead of the password itself reduces a risk of password leak, during a possible attack. However, after the authentication was successful, the server replaces the previously stored password with the one it received. Each consecutive authentication looks similar, where the user has to compute the password, by applying the hash function proper amount of time. In a situation that P reaches value of N, it means that all of the passwords have been used. Then, a new seed must be chosen, in order to generate a new set. Lamport's scheme has two main advantages:

- It brings the common advantage of one time passwords, that is resistant to replay attacks. An adversary capturing a genuine password is unable to use it. Additionally, instead of replaying the password one could try to generate valid keys for further usage. However, this would require breaking the pre-image resistance property of the used hash function, which is usually not feasible.

- The scheme is highly effective from the perspective of the server. It requires very limited space, as only one value has to be stored. What is more, a password leak does not give any advantage to the adversary. This is possible, because the data stored on the server is used only for password validation - it cannot be used to authenticate users.

### 4.2.2. Transport layer security

The Transport Layer Security (TLS) protocol [23] is one of the most commonly used real life solution for key transportation. It is the next generation of the Secure Socket Layer (SSL) protocol. The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. One of the main characteristics of the TLS is ability to enable communication between parties, that do not have to rely on the same cryptographic algorithms. In order to achieve that goal, the first phase involves a numerous negotiations between the communicating parties. They choose the most suitable algorithms, known to both entities, together with the version of protocol that should be used. After all of the components to be used are selected, parties exchange their certificates, to validate the authenticity. In the basic version of the protocol, it is common to authenticate only one side, usually the server a client tries to reach. In case of mutual authentication, an additional message has to be added. The key exchange itself is performed using a solution, that is usually based on asymmetric encryption techniques, for example Diffie-Hellman key exchange discussed in section 2.1. Integrity of messages can be achieved using message authentication codes.

### 4.2.3. Smart card authentication

A smart card is a small device, that includes an embedded integrated circuit called a smart chip. It contains a secure processor with internal memory, able to perform some computations. They may have up to 8 kilobytes of RAM, 346 kilobytes of ROM, 256 kilobytes of programmable ROM, and a 16-bit microprocessor [13]. The data stored on the card can be updated with information supplied by the terminal. Cards do not contain any power source. Instead, they are provided with the necessary energy by the card reader. Smart cards are easy to use. Therefore, they find application in various areas, such as banking transactions or authentication. The latter is the main concern for this thesis. An authentication process using smart cards may be achieved in multiple ways. In general, the most important factors determining the card protocol involve expected card cost, security level and adaptation to existing standards and laws. The first technique involves usage of symmetric encryption. The secret key embedded in each smart card should be unique, so that discovery of the key does not compromise the entire system. The system creates the diversified key from the combination of a system master key and a unique card characteristic, such as the microprocessor's serial number. The main weakness lays in the master key being stored on the card and the back-end while the terminal does not have it. It can cause serious security problems when reader must perform authentication without access to the server [67]. Another way to achieve authentication using smart cards is based on asymmetric encryption. This method overcomes the major weakness of having the master key exposed, but creates a problem of card forging. The asymmetric key pair could be generated by the attacker and used for the authentication process. To solve the issue, identity certificates should be issued by the system administrators and provided for each card. That way the verifier can be assured that a key pair is not forged. The authentication can also be done using zero-knowledge techniques. They are less vulnerable to key compromise than previously mentioned methods. The reader can deduce that the smart card possesses the secret accreditation without possessing any part of it. This is accomplished by the verifier issuing one or more challenges and the prover responding with an equal number of responses. The security level of this method increases exponentially with the number of challenges. However, it also increases the time, memory and energy consumption necessary for a single authentication procedure. Cards operating with zero-knowledge based protocols usually require cards with random number generators and exponentiation units, which increases the cost of the system [48]. A designer of a system should carefully pick the best underlying principle for the smart card authentication. As shown during the discussion, all of them come with a set of advantages and problems. Many smart card authentication schemes are vulnerable to the relay attack. There are two main solutions to the problem. The first one is a physical protection of the card. Introduction of a protected card holder prevents the adversary from establishing communication with the card. The second method is addition of Two Factor Authentication, for example using a password [37]. It has been shown that the latter mitigation can be overcome [1]. However in a situation of user authentication to a device they are going to use we consider this scenario not relevant.

## 4.3. Protocol

In this section we will describe the proposed scheme in details. Every parameter used is described in Table 4.1. For ease of understanding, we divide the protocol into 3 phases:

- *Phase 1: Deployment*

  This phase is performed on a manufacturing level of the components. Every implant $I$ is assigned an $ID_I$ and a secret, random 128 bit value $x$. That value is also stored in the server $S$, in a form of pairs of

| Symbol | Name of variable |
|---|---|
| $x, y$ | Established secret |
| $S$ | Server |
| $U$ | User of the reader |
| $I$ | Implant |
| $R$ | Reader |
| $ID_P$ | ID of the party P |
| $c_U, c_R$ | Challenges |
| $r_U, r_R$ | Responses |
| $\tau$ | Token |
| $T$ | Timestamp |
| $K$ | Session key |
| $MK$ | Master key |
| $N$ | Freshly generated nonce |
| $IDx$ | state ID of the hashchain |
| $PK_P$ | Public key of the party P |
| $SK_P$ | Private key of the party P |
| $h(M)$ | Hash function on data M |
| $h_1$, $h_2$ | Respectively the first and the second half of the digest |
| $E_{\mathscr{K}}(M)$ | Symmetric encryption using the key $\mathscr{K}$ on data M |
| $\mathscr{E}_{\mathscr{K}}(M)$ | Asymmetric encryption using the key $\mathscr{K}$ on data M |
| $D_{\mathscr{K}}(M)$ | Symmetric decryption using key $\mathscr{K}$ on enciphered data M |
| $\mathscr{D}_{\mathscr{K}}(M)$ | Asymmetric decryption using key $\mathscr{K}$ on enciphered data M |
| $Cert$ | Certificate of identity |
| $Sig_P(M)$ | Digital signature performed by party P on message M |
| $HMAC_{\mathscr{K}}(M)$ | Hash-based Message Authentication Code using key $\mathscr{K}$ on data M |

Table 4.1: Table of symbols

$ID_I$ and the corresponding seed $x$. In a similar manner, each reader $R$ is assigned $ID_R$ and each user $U$ authorized to use them is provided a smart card with $ID_U$. Due to our threat model, we have chosen user authentication based on asymmetric ciphers. This solution does not need any sensitive information to be embedded on the reader, which acts as the verifier. Additionally, the implementation costs and the message overhead are smaller than for the solutions based on the zero-knowledge techniques. Given that choice, each reader and smart card know $PK_S$. Additionally the smart cards are granted with a pair of $PK_U$ and $SK_U$. To prove that a card is valid, it also is embedded with a $Cert$ from the manufacturer, in a form of $Cert = (T, ID_U, PK_U, Sig_S(T|ID_U|PK_U)$. $T$ stands for time, until which the card should be considered genuine. If a card has been stolen and should be voided, the $ID_U$ is black-listed. Track of the list is kept by the server. A copy of the record of black-listed cards is kept on the reader, and updated by the server whenever necessary.

- *Phase 2: User Authorization*

  The second phase aims to establish the identity of the person trying to access the implant and identity

of the reader. Detailed layout of the protocol can be seen in Figure 4.2. The phase starts when the user puts the smart card near the reader. When detected, the reader sends the current time $T$, its certificate, a fresh challenge $r_R$ and a request to the card for its credentials. The card responds with its own certificate $Cert_U$ and a freshly generated challenge $c_U$. In the next step the reader computes a response:

$$r_U = \mathscr{E}_{SK_R}(c_U|c_R|T)$$

By binding both challenges together with $T$ we ensure that the responses have to be freshly computed and cannot be reused later. When the card receives $r_U$ it validates the received certificate $Cert_R$. It must check 3 things. Firstly it checks if the $ID_R$ is not on the list of readers known as malicious. Next, the card checks if the certificate was signed in a proper way, using $SK_S$. It can be checked using $PK_S$ embedded in the card. Lastly, the card inspects if the certificate has not expired. If the verification was successful, the card verifies $r_U$ using information retrieved from $Cert_R$. If is is considered genuine, the card computes $r_R$ in a similar manner as $r_U$ was computed and a token $\tau$, which will be used to bind the user authentication to the next phase. In the last step the reader does identical certificate verification and $r_R$ validation. By the end of this step we have assured that only authorized personnel is operating the reader. Additionally we have ensured that the communication is fresh and not reusable. Finally, we prevent the adversary from tricking any party to do a certificate verification. As discussed at the end of Section 4.2.3 the main threat during this phase is the relay attack. As a mitigation mechanism we suggest an enhanced security for the cards themselves, for example a case blocking any communication. This solution does not add any major issues and is more user friendly than introduction of two factor authentication.

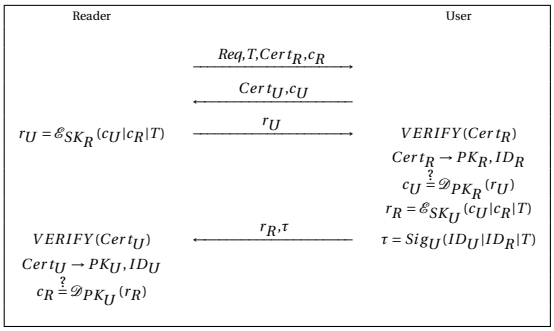



Figure 4.2: User authentication protocol

- *Phase 3: Key agreement*

  After the user was able to enable the reader, they can access the implant. The layout of the protocol can be seen in the Figure 4.3. The first step is creation of a secure channel between the reader and the server. In order to guarantee that the user has authenticated to the reader the server verifies correctness of the token $\tau$ using $PK_U$. After the secure tunnel has been established, the communication between the reader and the implant can start. Since the device is in sleep mode most of the time in order to save energy, the reader initiates the communication with a *Hello message*, which includes:

  $$ID_R|ID_U|\tau$$

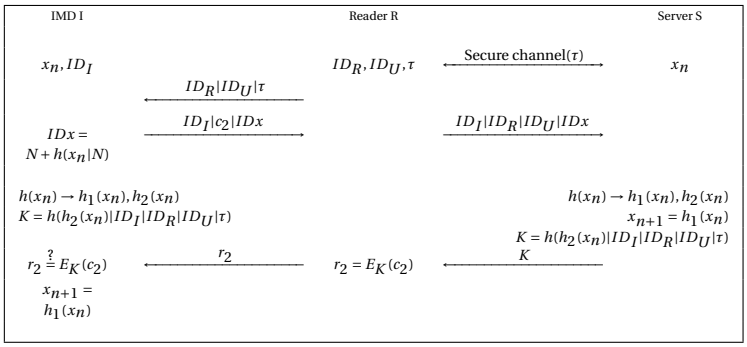



Figure 4.3: Key agreement protocol

The implant computes $IDx$ as follows:

$$IDx = N + h(x_n|N)$$

By the notion of $x_n$ we describe the value of $x$ in $n$-th execution of the protocol. The $IDx$ value is used to ensure that both the server and the implant always have the same state. The server can keep the initial value of $x$, denoted as $x_0$, and in case of synchronization problems can traverse the hash chain as long as $x_n$ is not obtained. In the next step the implant responds with a following message:

$$ID_I|c_2|IDx$$

At this stage both devices are in possession of IDs of every involved party. That information is sent to the server via the secure channel, together with a request for a session key $K$. Both server and implant compute the session key in a following way. $x_n$ is hashed producing 256 bit digest, which is later split into two halves, $h_1(x_n)$ and $h_2(x_n)$. In case of successful authentication the first 128 bits $h_1(x_n)$ will be used to create $x_{n+1}$, which will be used when another authentication attempt is done. Second half of the hash $h_2(x_n)$ is concatenated with the IDs of involved parties and hashed and $\tau$, which forms $K$:

$$K = h(h_2(x_n)|ID_I|ID_R|ID_U|\tau)$$

Due to complexity of asymmetric encryption the implant is unable to verify the correctness of the token $\tau$. However if the token is not genuine, the server will not provide $R$ with $h_2(x_n)$ and the secure session will not be established. When in possession of $K$, the reader can create a response for $c_2$, which takes a form of:

$$r_2 = E_K(c_2)$$

The implant can easily verify the correctness of $r_2$ by decrypting it and comparing with previously sent $c_2$. If it is correct, the ongoing communication is encrypted using $K$, ensuring its confidentiality. Additionally the value of $x$ can be updated. To ensure confidentiality and integrity of the communication, the messages are exchanged in a following format:

$$E_K(Message)|HMAC_K(E_K(Message))$$

It should be noted that the cryptographic primitive for $HMAC$ algorithm is a hash function. Therefore it does not bring additional complexity to the design of the implant.

After the protocol has been executed we have established a secure channel with the IMD. What is more, we have managed to establish that the user has authenticated to the reader. The presented protocol ensures the freshness of the communication. The session key is constructed using IDs of all participating parties. What is more, the validity of the user is ensured in the previous phase and tied together to the current session of the secure channel. What is more, we have designed the protocol in a way that leakage of information is reduced to the minimum. In the list below we present possible consequences of different malicious actions:

- *Eavesdropping* - adversary can obtain values of $ID_R, ID_U, ID_I, c_2, IDx, r_2$. Any other critical data is either encrypted or kept on the secure server. All of the IDs are public values, $c_2, r_2$ are generated freshly during each protocol execution. $IDx$ might appear as a big value for the attacker. By reverting the hash and obtaining $x_n$ one can decrypt all messages in the future, as well as make any number of authentications to the IMD. However, $x_n$ is protected by a secure one-way hash function. Additionally, even if the adversary is able to find a collision for $h(x_n|N)$ denoted as:

$$x_n \neq A$$
$$h(x_n|N) = h(A|N)$$

  it is unlikely that $A$ is an element from the hash chain:

$$h(x_n) \stackrel{?}{=} h(A)$$
$$h_1(x_n) \stackrel{?}{=} h_1(A)$$
$$h_2(x_n) \stackrel{?}{=} h_2(A)$$

  Because of the described observation and computational difficulty of reverting hash functions, this attack vector is not considered as a threat to the scheme.

- *Modification* - adversary can modify the values, causing the key agreement to fail. It has the same effect as communication channel blockage, which is out of scope of this work. Any possible desynchronization caused by this operation can be mitigated by $IDx$ value in further protocol executions.

- *Broken connection and session key theft* - adversary could break the connection and keep the valid session key $K$ until another run of the protocol is executed. To prevent this we propose a following mechanism. In case of no messages being sent, the implant sends a probe message in a form of $N$ and expects a response from the reader in form of $h(ID_I|ID_U|ID_R|N)$. Those messages are protected with encryption and HMAC as described above. Both broken connection (i.e. no probe message answer) or a proper end session message would notify the implant to discard $K$ and request another run of the key agreement protocol.

## 4.4. Emergency access

Environment of medical devices is different to many other existing fields. Since a lot of systems are responsible for keeping patients healthy, or in extreme cases alive, during the design the most emphasis is put onto the reliability. In other words, the most important is the ability of the system to perform its actions, rather than making it as secure as possible. Let us consider an example of online banking. If some resources are not available at a given time, the systems will be unavailable. Clients will have to wait, until the operations can be executed. This behavior is caused by the security measures. Resource denial that results in lowered security

level could be easily exploited by the adversary and would be a clear flaw of the system. The situation is different in case of medical systems. As described in Section 4.3, the proposed model requires constant connection to the Internet, in order to be able to connect to the secure server. It creates a possible target with very large value for the adversary. Being able to deny Internet connection, or performing a DoS attack on the server, would make the scheme unusable during an attack. Such pause could affect multiple lives, therefore is not acceptable. For this reason, we introduce an additional mechanism, that enables the access to the implant in any situation, regardless of accessibility of the Internet.

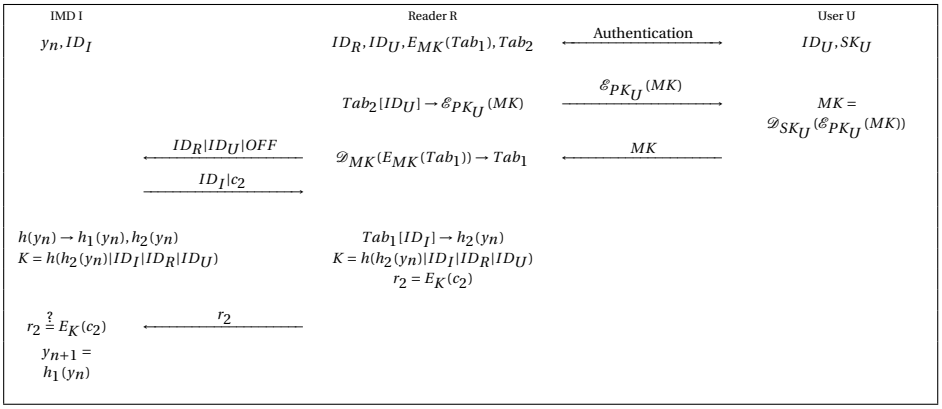



Figure 4.4: Offline key agreement protocol

The details of the offline mode are shown in Figure 4.4. On the manufacturing level, the implant is provided with a second secret $y_n$, that is used to create session keys for offline mode of operation. In the discussed scenario the server is not accessible, therefore reader must obtain the keys in some other way. For this purpose, the reader is additionally embedded with two data tables. The first table $Tab_1$ consists of multiple tuples of $ID_I$ and $h_2(y)$, and $ID_y$ in form of $ID_y = h(ID_I|h_2(y)$. The first two columns of the table are encrypted using a symmetric cipher using $MK$. It is a key created during the deployment of the system by the system manufacturer. It protects the material used to generate session keys for the offline mode. Therefore it is kept secret on the secure server and not shared with any other party. The second table $Tab_2$ contains pairs of $h(ID_U)$ and $\mathscr{E}_{PK_U}(MK)$. Whenever a new $U$ should be added, the server computes a needed pair and updates $Tab_2$ on each reader. After the user authorization phase, one must switch to the offline mode. By performing this operation, the reader will find the $ID_U$ in $Tab_2$ and corresponding cipher text $\mathscr{E}_{PK_U}(MK)$. It will be sent to the card, which will perform the decryption using $SK_U$, to provide the reader with $MK$. The the reader initiates the connection to the implant in the same way as in the Key agreement phase described above, with specifying that it intends to use the offline mode using the $OFF$ flag. Except for connecting the server, the reader obtains the needed $h_2(y_n)$ in the $Tab_1$ and follows the rest of the protocol in the regular way. Procedures for the implant do not change, the only difference is second hash chain that must be used for key generation.

With the proposed solution the doctors would have only one try at establishing the connection to the IMD. Therefore to increase reliability, we introduce additional protection measure. After the session key in the offline mode has been used, the implant begins a countdown for a period of time $t$. If anyone wants to make a consecutive connection without the server, it must be done before the countdown finishes using the same session key. Successful authentication using the same key stops the current timer, and start a new

countdown for the same amount of time $t$. That way, the doctors must keep track of time, in order to refresh the timer and keep the key valid. We acknowledge that this solution does not grant the perfect availability. However, we assume that providing the patient with a second treatment in the time window $t$, or finding a way to establish connection with the server is sufficient.

After the session key is used, we update the $ID_y$ with the new key $h_2(y_{n+1})$. This value is checked every time a connection is made to the implant in online mode. That way the paramedics are able to track if the offline mode has been used for current IMD. If that is the case a request is send to the server in order to update $Tab_1$. We consider the detection in offline mode unrealistic, as doctors are aware that subsequent connections must be made in the time window created by $t$.

The main threat to the offline mode of operation is theft of the session keys embedded in the reader. Therefore, they should be stored in a safe memory and properly encrypted using strong AES cipher with 256bit key to prevent decryption. In this mode we are not using state ID, since the reader is embedded with only one session key. To prevent desynchronization we ensure that the value of $y$ is updated only after successful authentication attempt. The rest of the aspects of the protocol are the same as for normal mode of operation, therefore the analysis performed in the previous section applies here as well.

### 4.4.1. Offline mode alternatives

The offline mode is one of the weakest links in the proposed protocol. Adversary that is able to steal the reader and a valid authentication card is able to gain access to the one time keys for any device. In order to increase the difficulty of this attack, we propose requirement of multiple users authenticating to retrieve the keys. This can be achieved by introducing Shamir's secret sharing [61]. Each smart card is encoded with a secret share - in this case a point laying on a polynomial. When the special mode is needed, the parties provide the reader with their shares, which enable the master key reconstruction, using the Lagrange interpolation, as described in Section 3.3. With this mechanism, the adversary is forced to possess the reader and multiple smart cards, before the system is notified about theft and the cards are invalidated. The perks of this solution is ease of customization. Based on the customer needs, the minimum number of users can be easily adjusted. We have not introduced this idea to the main version of the protocol, since it might be at the cost of a large reduction of system reliability. Every connection during the offline mode would require multiple authorized personnel. In the case of an emergency, paramedics have large amount of work, having multiple of them gather to enable communication with the IMD might be impossible.

It might be the case, that even the basic version of the proposed offline mode is considered not sufficiently reliable. As an alternative, we recommend implementation of the Rasmussen's ultrasonic distance bounding protocol [55]. It was shown in the Section 3.2, that the protocol does not contain any major known vulnerabilities. It should be noted, that the replacement will most likely increase the implementation cost of both the implants and the readers, as they have to be equipped with ultrasonic communication interfaces that have to be properly shielded.

# 5

# Protocol analysis

In the previous chapter we have have described a set of concise requirements that the proposed protocol should fulfill. With the description of the protocol we have shown, that the data stored on the reader, as well as data transmitted during the authentication phase does not leak any sensitive information. What is more, we have shown an emergency mode for our system, when there is no Internet connectivity for the reader. In the following section we will evaluate whether the introduced scheme satisfies the low energy consumption requirement. It is highly important for both usability and security. Low energy consumption is necessary for any operations performed on the IMD as the battery capacity determines the lifespan of the implant. Security wise, proper energy management helps to mitigate serious threats, such as the battery DoS attack.

In order to enable the implementation of the protocol into a real life solution, in this section we present all of the necessary technical details. We provide a discussion of advantages and disadvantages of particular choices. Later, we show the results of the energy measurements of particular cryptographic primitives. Based on the choice, we provide a resource consumption summary, including necessary energy and static memory for the protocol to operate.

## 5.1. Protocol implementation

Due to the limited power of the implants, it is important to find proper balance between security and efficiency. Our proposed solution requires two main primitives: a block cipher and a hash function. We have implemented different algorithms, in order to measure their efficiency. For the purpose of the implementation, ARM Cortex-M4 32-bit microcontroller for Silicon Labs was used (EFM32 Giant Gecko Series 1) [39] The main reasons behind the choice are the MCU's low power consumption that illustrates well the IMD environment, and increasing popularity of the MCU in the IoT devices. To measure the amount of used resources, we used the Advanced Energy Monitoring and Simplicity Studio software [40]. The compilation was done without any compiler optimisations (*-O0* flag). The MCU is running in its active mode under current of 2.03mA and voltage of 3.3V. Those values are used for energy consumption calculations. We calculate the consumed energy as a product of voltage, current and time. The time is calculated using clock cycles needed to execute a given algorithm. Number of cycles is constant for a given algorithm, despite the varying input data. With voltage being constant as well, the only fluctuating this is the current. In the calculations we use the average

reading. We have observed fluctuation of current of approximately 4%. The tests consisted of the necessary operation needed for a single authentication using the protocol, that is double hashing and memory operation to create the session key. We have chosen five different hash algorithms for comparison. SHA-1 [26], SHA-256 [53] and SHA-3 [10] are the representatives of the Secure Hash Algorithm group. Photon [30] and Quark [4] are specially designed to be lightweight hash functions, suitable for implementation in restricted environments[1]. The results of the hash function comparison can be seen in Table 5.1. After comparing the results, it was discovered that the SHA-1 and SHA-256 were the most efficient. There are a few possibilities why the lightweight algorithms were more resource consuming. The first one is that they were designed to be implemented in hardware, rather than software. Manufacturing specially designed circuits for the purpose of the comparison was not feasible, therefore we have considered only the C code implementations. The second possibility is that the tested code was not optimized. After further improvements it might have been possible to achieve better results. Since the SHA-1 was found to be vulnerable to collision attack [64], we suggest using SHA-256 for implementation of the protocol.

The same tests were performed on the block ciphers. In this case, we have tested encryption of 128 bits of data. This is the size of the largest block size for the chosen ciphers. Therefore, it will make the calculations for any message size easier. As far as the sizes of messages to be encrypted are concerned, we have not designed the content of the messages, that is: queries and replies. The only message of fixed size is the challenge $c$, where the length is 256 bits. We have selected four different algorithms: the Advances Encryption Standard (AES) [18], Misty1 [45], Simon and Speck[6][2]. The latter three were designed to be lightweight block ciphers. It should be noted that due to different block sizes, AES cipher had to encrypt a single block of data, while the rest had to encrypt two blocks. The results of the test can be seen in Table 5.2. Clearly the AES cipher requires much larger resources amount than the other test subjects. Despite good performance energy wise, the Misty1 algorithm was found vulnerable to an attack which significantly reduces the key size [51] and is not a preferable choice. Simon and Speck are one family of ciphers. The first one was designed to be implemented in hardware, the latter in software. As far as their security goes, multiple papers with cryptanalysis were proposed. As of the time of writing this thesis, no successful attack on full-round Speck of any variant is known, while the reduced-round variants have been successfully attacked [7]. For this reason we suggest picking one representative of the family. If the cipher can be implemented in hardware, we propose the Simon cipher. If it is to be included as a software component, we advise the Speck cipher.

[1] The C implementations of hash functions were obtained from:

- SHA-1: https://github.com/clibs/sha1
- SHA-256: https://github.com/B-Con/crypto-algorithms
- SHA-3: https://github.com/rhash/RHash/tree/master/librhash
- Photon: https://sites.google.com/site/photonhashfunction/software
- Quark: https://github.com/veorq/Quar19920k

[2] The C implementations of ciphers were obtained from:

- AES: https://github.com/kokke/tiny-AES-c/blob/master/test.c
- Misty1: https://github.com/stamparm/cryptospecs/blob/master/symmetrical/sources
- Simon and Speck: https://github.com/inmcm/Simon_Speck_Ciphers

| Hash function | Clock cycles | Time of execution (ms) | Total energy ($\mu$J ) | Static memory (bytes) | Flash memory (bytes) |
|---|---|---|---|---|---|
| SHA-1 | 27476 | 1.446 | 9.687 | 188 | 18896 |
| SHA-256 | 30375 | 1.599 | 10.71 | 188 | 14928 |
| SHA-3 | 145460 | 7.656 | 86.449 | 380 | 23116 |
| Photon | 32257021 | 1697.738 | 11373.15 | 456 | 21700 |
| Quark | 57701267 | 3036.909 | 20344.25 | 220 | 20084 |

Table 5.1: Hash functions comparison

| Cipher | Block size (bits) | Key size (bits) | Clock cycles | Time of execution (ms) | Total energy ($\mu$J ) | Static memory (bytes) | Flash memory (bytes) |
|---|---|---|---|---|---|---|---|
| AES | 128 | 128 | 31714 | 1.669 | 11.182 | 188 | 14660 |
| Speck | 64 | 128 | 12354 | 0.65 | 4.356 | 168 | 16244 |
| Misty1 | 64 | 128 | 9958 | 0.524 | 3.511 | 2344 | 15796 |
| Simon | 64 | 128 | 15970 | 0.841 | 5.631 | 208 | 16956 |

Table 5.2: Block cipher comparison

## 5.2. Resource consumption

In the following section we will perform an estimation of resources required to execute the protocol on the IMD. In the beginning we will analyze how much energy is required for a single protocol run. Secondly, we will calculate the amount of memory needed. This is an important parameter when considering implementation of the solution, as it dictates what are the mandatory parameters of the chosen hardware components. During the calculation we will use values presented in the previous section for SHA-256 and the Speck cipher.

### 5.2.1. Energy usage

| Step number | Step | Time of execution (ms) | Total energy ($\mu$J) |
|---|---|---|---|
| 1 | Receive $ID_R \vert ID_U$ | 0.32 | 5.25 |
| 2 | Generate $c$ | 0.61 | 4.025 |
| 3 | Sleep/Sniff | 0.61 | 0.011 |
| 4 | Send $ID_I \vert c$ | 0.8 | 14.429 |
| 5 | Compute $K$ | 1.599 | 10.549 |
| 6 | Sleep/Sniff | 500 | 9.395 |
| 7 | Receive $r_2$ | 0.64 | 10.342 |
| 8 | Decrypt $r_2$ | 0.65 | 4.289 |

Table 5.3: Energy usage in a single execution of the protocol

In this subsection we will demonstrate how much energy is necessary for each step of the protocol shown in the previous chapter. In addition to the required cryptographic operations, we are going to take into account the amount of energy consumed by the transceiver. For this purpose we are going to use ZL70103 Medical Implantable RF Transceiver [46]. It is a high-performance, half-duplex, RF communications link for medical implantable applications. One of the main reasons they are used in IMDs is their reliability, which is crucial when designing a critical systems. For the operation it has data-rate of 400kb per second, uses voltage of 3.3V, and the currents of 4.3mA and 4.8mA for reception and transmission respectively. Between those modes, the device is sleep mode and periodically goes to sniff mode - the average current needed for these is 5 $\mu$A. The overview of the table can be seen in Table 5.3. To further highlight the energy costs, we have grouped the operations into 3 categories. Figure 5.1 shows how much energy each group consumes for the computations and for the communication. In the measurements we assume that the challenge $c$ is 256-bit value, generated using the hash function. The size of the IDs used in the protocol is 64 bits. Additionally we assume that the amount of time needed for the reader to contact the server and get a reply equals 500ms[3]. With those assumptions, the total energy consumed by a single authentication process equals 58.248$\mu$A over a period of 105.629ms. As mentioned in Section 5.1, we have used the average readings from our measurements. It should be noted that the mentioned fluctuations of approximately 4% during execution of hash algorithms or block ciphers are very insignificant when compared to the costs of data transmission. As shown in Section 4.4, the emergency access does not differ from the perspective of the IMD. It has to receive additional flag in the first step, which could be as small as 1 bit. However, the time of the transceiver waiting in step 6 will be decreased, as there will be no communication with the sever. The calculations prove that the protocol is a lightweight solution, which will affect neither the performance nor the lifetime of the IMDs.

[3] It is difficult to determine precisely how much time will be needed for the communication with the server. It can be affected by multiple factors, such as connection quality at a given place, server workload at a given time etc.

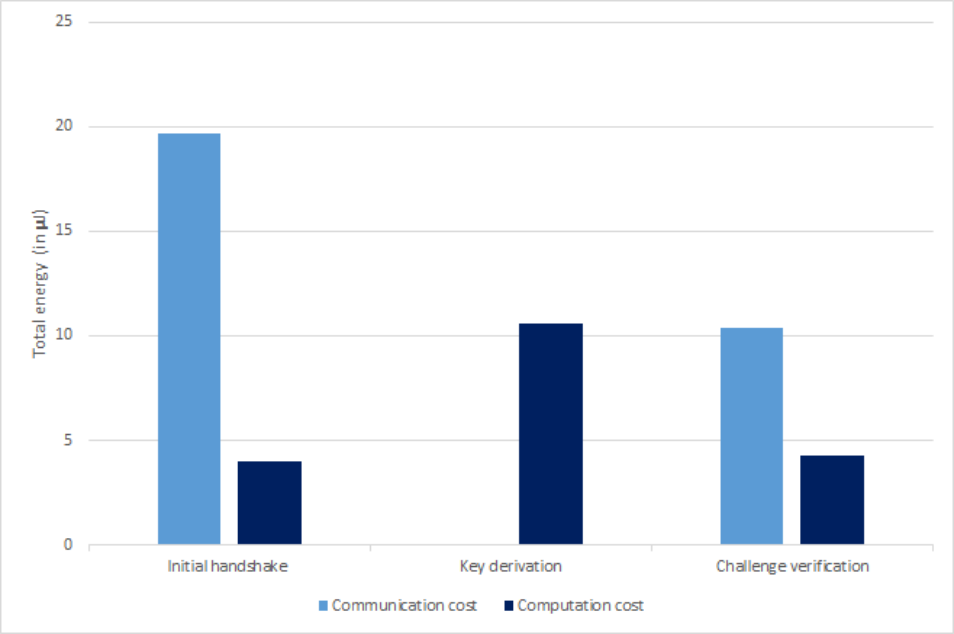



Figure 5.1: The energy costs of the protocol

The energy usage was computed only from the perspective of the implant. Since it is a resource constrained device. Other components in the proposed model do not have to follow such strict regulations. However, there are other considerations that should be taken into account. In case of the server, the time and computational power necessary for an individual process are important factors. The main reason behind it is the fact that the server must be able to execute multiple transactions in parallel. While a single process might be lightweight and would not cause any problems, having hundreds or thousands of them at the same time could cause disruptions. In our scheme, the server performs the same computations as the implant. Due to the natural limitations introduced by a small embedded device it can be easily noticed that the process does not require any significant resources. Because of that, parallelization of the work can be executed smoothly, without excessive funds spent on the hardware. As far as the reader is concerned, it is the least limited component. It has virtually unlimited power as it can be plugged into a power source. It will execute the protocol with at most one IMD at a time, removing the problems of aligned tasks. However, despite the mentioned statements, the reader has a low amount of instructions to be executed in our scheme. One could reason that it is a wasted opportunity not to allocate any computations to this device. However, in our threat model the reader is not a trusted component. Therefore, in the protocol we tried to limit its role, making it less attractive target for the adversary.

### 5.2.2. Memory usage

When designing the protocol we tried to keep in mind that some components may have limited storage capabilities. While it is a clear assumption regarding the implant, one could also think about the other components. In case of the reader, large necessary storage could increase the manufacturing cost, since the devices are going to be produced in bigger quantities. In case of the server, just a small increase in the amount of data that has to be stored per user could lead to considerable escalation, as there are numerous users served by the machine. The amount of necessary memory for each device is shown in Table 5.4. The implant requires almost no memory for the data related to security protocol. Since the session keys are generated freshly from

| Component | Entry | Size | Number of entries | Total size |
|---|---|---|---|---|
| Implant | Embedded secrets | $ID_I$ - 64b, $x$ - 128b, $y$ - 128b | 1 | 40B |
| Reader | Table 1 | $ID_I$ - 64b, $h(y)_2$ - 128b | 100 000 000 | 2.4GB |
| Reader | Table 2 | $h(ID_U)$ - 256, $E_{PK_U}(MK)$ - 2048b | 100 | 28.8kB |
| Server | Information per user | $ID_I$ - 64b, $x$ - 128b, $y$ - 128b | 100 000 000 | 4GB |

Table 5.4: Memory usage per component

a small seed, the scheme does not affect the implant manufacturers in a meaningful way. The reader has to keep the information used for the emergency mode for every implant it could encounter, while being operated by any of the possible paramedics. Given an estimate of 100 000 000 implants and 100 paramedics per device, each reader would use almost 2.5GB of memory. This is a very low value for the technology available nowadays. In case of the server, the most memory is occupied by the secrets $x$ for regular key establishment and $y$ for the emergency mode with the corresponding implant ID. Using the same estimate of existing implants, we obtain only 4GB of data. This makes the server easily expandable by additional implants, or other necessary data.

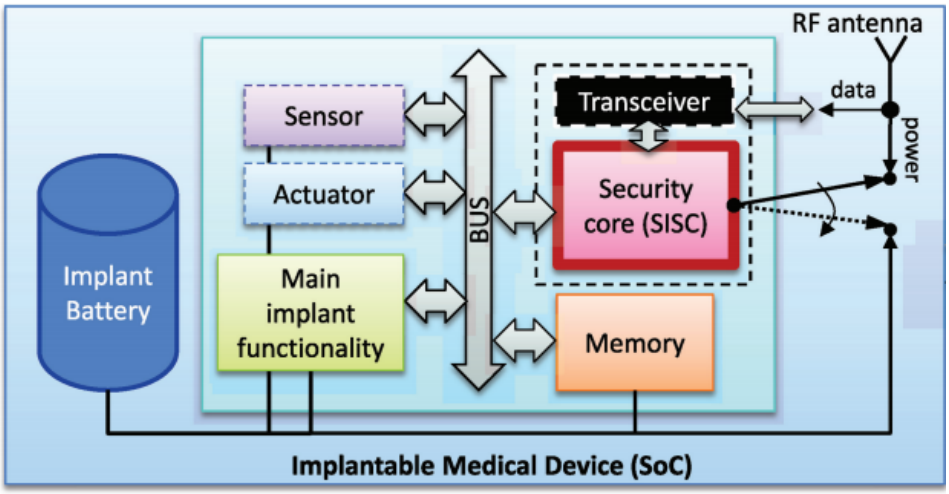



Figure 5.2: System architecture presented by Strydis et al.[65]

## 5.3. Prevention of battery depletion

Many researchers when proposing new protocols do not focus on battery depletion attack. Since the batteries are usually not rechargeable, it creates a significant opportunity for the adversary. The basic version of the proposed protocol is not fully secure versus this kind of an attack. The adversary is able to send multiple authentication requests to the IMD. Each of them would have to be verified. The costs of the verification are not major, but when multiplied by number of requests, this attack can significantly decrease the battery life of the implant. To fully mitigate the issue, we introduce a solution proposed by Strydis et al. [65]. The main idea of the solution is to split the hardware application, as shown in Figure 5.2. All of the modules necessary for the implant to provide the needed functionality are powered by the internal battery. Any security and authentication related tasks are performed by the Smart-Implant Security Core (SISC). The module is powered additionally by the energy harvested from the reader through the RF antenna. Any communication

attempt is started by waking up the SISC through the transceiver module. The authentication step is fully powered by the harvested energy. Therefore, if the communication request is an attack, the implant is able to drop the connection without discharging the battery. In effect, the adversary may send unlimited requests without causing any damage. The authors have performed a set of tests on the efficiency of harvesting. They have proposed a novel authentication protocol, which consumed in total 7.45$\mu$J of harvested energy. Additionally, Seepers [59] presented a protocol based on the same concept, which consumes 88 $\mu$J of energy per session. That amount was considered viable for the proposed approach. When compared to the protocol introduced in this thesis, our protocol needs approximately 50.732$\mu$J per authentication operation. Slightly larger amount of energy could result in a small delay, as the time needed to harvest sufficient amount of energy would increase. It can be deduced that our protocol can potentially support a zero-power defense scheme, which is fully secure against the battery depletion attack.

# 6

# Conclusion

Implantable Medical Devices are an important part of the modern medicine. Due to increasing popularity of this treatment, it became vital to ensure that the implants do not pose any additional threats to the patient. While the medical implications are dependent on the doctors, cyber threats are also an important factor. Because of that, the researchers are trying to create proper authentication schemes for the devices to ensure that the privacy and security of the patient is not violated.

The goal of this thesis was to make a contribution to the field of securing the Implantable Medical Devices. We have proposed a novel security protocol, that allows mutual authentication of the implants and the readers, as well as establishment of a common, freshly generated session key for any communication. Additionally, we have ensured that the proposed scheme meets the requirements of the strict environment of the IMDs by choosing operations and primitives that do not require computationally expensive tasks.

In the beginning of the thesis we have stated a couple of research question. By the means of discussion of the related work in the field and analysis of the obtained results, we have managed to answer all of them, while presenting explanation and possible limitations. Firstly, we have investigated the current state of the art security protocols. By inspecting solutions based on different underlying principles, such as usage of physiological signals or distance bounding, we have managed to create a comprehensive comparison of their advantages and drawbacks.

Secondly, we have performed an extensive security analysis of chosen protocols. We have summarized existing vulnerabilities discovered by fellow researchers. Additionally, we presented a new attack on a physiological-signal based key agreement protocol. By exploiting the main cryptographic primitive, which is the fuzzy vault, and the nature of the ECG signal, we have managed to create an attack, that is easy to perform and has a high probability of success reaching 70%.

Given the discovery of major vulnerabilities in the existing protocols, we have created a novel security protocol. Knowing the limitations of the implants, we have chosen the hash chains to be the main cryptographic primitive used. One of the largest accomplishments of the new solution is coverage of an additional attack vector - reader theft. We have managed to fulfill the requirement by introducing a concept that was not widely adopted, which is user authentication. In our scheme the personnel using the reader must undergo an authentication procedure. It increases the overall security by limiting the access to the IMDs, while

enabling tracking of any modifications to the device with potentially harmful impact. The additional attack vector poses a large limitation in the design, as no sensitive information can be stored on the reader. We have overcame that restraint by using encryption on all of the important data, together with smart key management between authorized personnel. Due to specific nature of the IMDs, we have introduced a fallback authentication mechanism to ensure that the patients can be treated in all circumstances. Finally, we have performed a large set of efficiency tests, to ensure that the solution is suitable for IMDs. We have compared different cryptographic primitives in order to achieve desired levels of efficiency, without major sacrifices of usability or security.

As far as the future work is concerned, there are various possibilities. The modularity of the proposed system gives a lot of freedom on how it can be adapted to satisfy different needs. We encourage further studies about IMD security to develop protocols that do not require pre-shared common secrets, while being free of the problems existing in current solutions. That way the emergency solution proposed in our work could be enhanced, to reduce the amount of data stored on the readers even further.

Overall, the thesis is a noticeable improvement to the IMD security standards. The major contributions include:

- **Comparison of existing protocols** - we have investigated different solutions proposed by other researchers and compared their benefits and drawbacks.
- **Implementation of a novel attack on the state of the art protocol** - we have demonstrated that it is possible to break an existing security solution
- **Proposition and evaluation of a new security scheme** - we have proposed a novel security scheme that includes authentication of the personnel

We believe that the presented work will enhance the quality of subsequently proposed solutions. We think that other researchers should consider addition of personnel authentication to their schemes, as it brings various benefits without major sacrifices. Additionally, being aware of a different attack type should result in better analysis of the proposed work, leading to more secure protocols.

# Bibliography


[1] Ben Adida, Mike Bond, Jolyon Clulow, Amerson Lin, Steven Murdoch, Ross Anderson, and Ron Rivest. Phish and chips. In *International Workshop on Security Protocols*, pages 40–48. Springer, 2006.

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

[3] Alessandro Armando, David Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, P Hankes Drielsma, Pierre-Cyrille Héam, Olga Kouchnarenko, Jacopo Mantovani, et al. The avispa tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification*, pages 281–285. Springer, 2005.

[4] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. Quark: A lightweight hash. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 1–15. Springer, 2010.

[5] Priyanka Bagade, Ayan Banerjee, Joseph Milazzo, and Sandeep KS Gupta. Protect your bsn: No handshakes, just namaste! In *Body Sensor Networks (BSN), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[6] Ray Beaulieu, Stefan Treatman-Clark, Douglas Shors, Bryan Weeks, Jason Smith, and Louis Wingers. The simon and speck lightweight block ciphers. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–6. IEEE, 2015.

[7] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. Notes on the design and analysis of simon and speck. Technical report, IACR Cryptology ePrint Archive, 2017.

[8] Tom Beer. Walsh transforms. *American Journal of Physics*, 49(5):466–472, 1981.

[9] Steven M Bellovin and Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *Research in Security and Privacy, 1992. Proceedings., 1992 IEEE Computer Society Symposium on*, pages 72–84. IEEE, 1992.

[10] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak sponge function family main document. *Submission to NIST (Round 2)*, 3(30), 2009.

[11] Carlo Blundo, Alfredo De Santis, Amir Herzberg, Shay Kutten, Ugo Vaccaro, and Moti Yung. Perfectly-secure key distribution for dynamic conferences. In *Annual International Cryptology Conference*, pages 471–486. Springer, 1992.

[12] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 450–466. Springer, 2007.

[13] Thomas Bronack. How do smart cards work.

[14] Haotian Chi, Longfei Wu, Xiaojiang Du, Qiang Zeng, and Paul Ratazzi. e-safe: Secure, efficient and forensics-enabled access to implantable medical devices. *arXiv preprint arXiv:1804.02447*, 2018.

[15] Wonsuk Choi, In Seok Kim, and Dong Hoon Lee. E2pka: An energy-efficient and pv-based key agreement scheme for body area networks. *Wireless Personal Communications*, 97(1):977–998, 2017.

[16] Jolyon Clulow, Gerhard P Hancke, Markus G Kuhn, and Tyler Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In *European Workshop on Security in Ad-hoc and Sensor Networks*, pages 83–97. Springer, 2006.

[17] Cas Cremers, Kasper B Rasmussen, Benedikt Schmidt, and Srdjan Capkun. Distance hijacking attacks on distance bounding protocols. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 113–127. IEEE, 2012.

[18] Joan Daemen and Vincent Rijmen. Aes proposal: Rijndael. 1999.

[19] Christophe De Canniere, Orr Dunkelman, and Miroslav Knežević. Katan and ktantan—a family of small and efficient hardware-oriented block ciphers. In *Cryptographic Hardware and Embedded Systems-CHES 2009*, pages 272–288. Springer, 2009.

[20] Gerard De Haan and Vincent Jeanne. Robust pulse rate from chrominance-based rppg. *IEEE Transactions on Biomedical Engineering*, 60(10):2878–2886, 2013.

[21] Tamara Denning, Kevin Fu, and Tadayoshi Kohno. Absence makes the heart grow fonder: New directions for implantable medical device security. In *HotSec*, 2008.

[22] Tamara Denning, Alan Borning, Batya Friedman, Brian T Gill, Tadayoshi Kohno, and William H Maisel. Patients, pacemakers, and implantable defibrillators: Human values and security for wireless implantable medical devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 917–926. ACM, 2010.

[23] Tim Dierks and Eric Rescorla. The transport layer security (tls) protocol version 1.2. Technical report, 2008.

[24] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[25] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[26] Donald Eastlake and Paul Jones. Us secure hash algorithm 1 (sha1), 2001.

[27] Gray Frank. Pulse code communication, 1953. US Patent 2,632,058.

[28] Shyamnath Gollakota, Haitham Hassanieh, Benjamin Ransford, Dina Katabi, and Kevin Fu. They can hear your heartbeats: non-invasive security for implantable medical devices. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 2–13. ACM, 2011.

[29] Lorena González-Manzano, José M de Fuentes, Pedro Peris-Lopez, and C Camara. Encryption by heart (ebh)—using ecg for time-invariant symmetric key generation. *Future Generation Computer Systems*, 77:136–148, 2017.

[30] Jian Guo, Thomas Peyrin, and Axel Poschmann. The photon family of lightweight hash functions. In *Annual Cryptology Conference*, pages 222–239. Springer, 2011.

[31] Daniel Halperin, Thomas S Heydt-Benjamin, Benjamin Ransford, Shane S Clark, Benessa Defend, Will Morgan, Kevin Fu, Tadayoshi Kohno, and William H Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 129–142. IEEE, 2008.

[32] Xiali Hei, Xiaojiang Du, Jie Wu, and Fei Hu. Defending resource depletion attacks on implantable medical devices. In *Global telecommunications conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[33] Saied Hosseini-Khayat. A lightweight security protocol for ultra-low power asic implementation for wireless implantable medical devices. In *Medical Information & Communication Technology (ISMICT), 2011 5th International Symposium on*, pages 6–9. IEEE, 2011.

[34] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006.

[35] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36. ACM, 1999.

[36] Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[37] Ziv Kfir and Avishai Wool. Picking virtual pockets using relay attacks on contactless smartcard. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 47–58. IEEE, 2005.

[38] Younghyun Kim, Woo Suk Lee, Vijay Raghunathan, Niraj K Jha, and Anand Raghunathan. Vibration-based secure side channel for medical devices. In *Proceedings of the 52nd Annual Design Automation Conference*, page 32. ACM, 2015.

[39] Silicon Labs. Efm32 giant gecko series 1 32-bit microcontroller, Accessed: 2018-10-15. URL `EFM32GiantGeckoSeries132-bitMicrocontroller`.

[40] Silicon Labs. Efm32™ giant gecko gg11 starter kit, Accessed: 2018-10-15. URL `https://www.silabs.com/products/development-tools/mcu/32-bit/efm32-giant-gecko-gg11-starter-kit`.

[41] Maryam Lafkih, Patrick Lacharme, Christophe Rosenberger, Mounia Mikram, Sanaa Ghouzali, Mohammed El Haziti, and Driss Aboutajdine. Vulnerabilities of fuzzy vault schemes using biometric data with traces. In *International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2015.

[42] Leslie Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.

[43] Chunxiao Li, Anand Raghunathan, and Niraj K Jha. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In *e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on*, pages 150–156. IEEE, 2011.

[44] Eduard Marin, Dave Singelée, and Bart Preneel. A survey on physiological-signal-based security for medical devices.

[45] Mitsuru Matsui. New block encryption algorithm misty. In *International Workshop on Fast Software Encryption*, pages 54–68. Springer, 1997.

[46] MICROSEMI. Zl70103 - wireless for implantable medical devices, Accessed: 2018-10-12. URL `https://www.microsemi.com/product-directory/implantable-medical-transceivers/3915-zl70103`.

[47] Dryden P Morse, Ugo F Tester, and Gerald M Lemole. The actual lifespan of pacemakers. *Chest*, 64(4): 454–458, 1973.

[48] RA Nelson. Authentication techniques for smart cards. Technical report, Westinghouse Hanford Co., 1994.

[49] B Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications magazine*, 32(9):33–38, 1994.

[50] Jerzy Neyman and Egon S Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Phil. Trans. R. Soc. Lond. A*, 231(694-706):289–337, 1933.

[51] A Bar On and N Keller. A 270 attack on the full misty1. In *Proc. 36th Annual International Cryptology Conference (CRYPTO 2016)*, pages 435–456, 2016.

[52] Jiapu Pan and Willis J Tompkins. A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3):230–236, 1985.

[53] Wouter Penard and Tim van Werkhoven. On the secure hash algorithm family. *Cryptography in Context*, pages 1–18, 2008.

[54] Steffen Peter, Bhanu Pratap Reddy, Farshad Momtaz, and Tony Givargis. Design of secure ecg-based biometric authentication in body area sensor networks. *Sensors*, 16(4):570, 2016.

[55] Kasper Bonne Rasmussen, Claude Castelluccia, Thomas S Heydt-Benjamin, and Srdjan Capkun. Proximity-based access control for implantable medical devices. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 410–419. ACM, 2009.

[56] Masoud Rostami, Ari Juels, and Farinaz Koushanfar. Heart-to-heart (h2h): authentication for implanted medical devices. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1099–1112. ACM, 2013.

[57] Michael Rushanan, Aviel D Rubin, Denis Foo Kune, and Colleen M Swanson. Sok: Security and privacy in implantable medical devices and body area networks. In *2014 IEEE Symposium on Security and Privacy (SP)*, pages 524–539. IEEE, 2014.

[58] Nitesh Saxena, Md Borhan Uddin, Jonathan Voris, and N Asokan. Vibrate-to-unlock: Mobile phone assisted user authentication to multiple personal rfid tags. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 181–188. IEEE, 2011.

[59] Robert Seepers. *Implantable Medical Devices : Device security and emergency access*. PhD thesis, E, December 2016. URL http://hdl.handle.net/1765/94389.

[60] Robert M Seepers, Jos H Weber, Zekeriya Erkin, Ioannis Sourdis, and Christos Strydis. Secure key-exchange protocol for implants using heartbeats. In *Proceedings of the ACM International Conference on Computing Frontiers*, pages 119–126. ACM, 2016.

[61] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[62] International standard ISO/IEC 9798-2. Information technology - security techniques - entity authentication, Accessed: 2018-26-06. URL bcc.portal.gov.bd/sites/default/.../ISO_IEC_9798-2.pdf.

[63] International standard ISO/IEC 9798-3. Information technology - security techniques - entity authentication, Accessed: 2018-26-06. URL https://www.sis.se/api/document/preview/912341/.

[64] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full sha-1. In *Annual International Cryptology Conference*, pages 570–596. Springer, 2017.

[65] Christos Strydis, Robert M Seepers, Pedro Peris-Lopez, Dimitrios Siskos, and Ioannis Sourdis. A system architecture, processor, and communication protocol for secure implants. *ACM Transactions on Architecture and Code Optimization (TACO)*, 10(4):57, 2013.

[66] Lisa Vaas. Doctors disabled wireless in dick cheney's pacemaker to thwart hacking, Accessed: 2018-03-09. URL https://nakedsecurity.sophos.com/2013/10/22/doctors-disabled-wireless-in-dick-cheneys-pacemaker-to-thwart-hacking/.

[67] Jordi van den Breekel, Diego A Ortiz-Yepes, Erik Poll, and Joeri de Ruiter. Emv in a nutshell. *Technical Report*, 2016.

[68] Krishna K Venkatasubramanian, Ayan Banerjee, and Sandeep Kumar S Gupta. Pska: Usable and secure key agreement scheme for body area networks. *IEEE Transactions on Information Technology in Biomedicine*, 14(1):60–68, 2010.

[69] Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, Mauro Conti, and Athanasios V Vasilakos. A novel authentication and key agreement scheme for implantable medical devices deployment. *IEEE journal of biomedical and health informatics*, 2017.

[70] Longfei Wu, Xiaojiang Du, Mohsen Guizani, and Amr Mohamed. Access control schemes for implantable medical devices: A survey. *IEEE Internet of Things Journal*, 4(5):1272–1283, 2017.