A survey on physiological-signal-based security for medical devices

Eduard Marin, KU Leuven, ESAT-COSIC and iMinds,
Enrique Argones Rúa, KU Leuven, ESAT-COSIC and iMinds,
Dave Singelée, KU Leuven, ESAT-COSIC and iMinds,
Bart Preneel, KU Leuven, ESAT-COSIC and iMinds,

Implantable Medical Devices (IMDs) are used to monitor and control patients with chronic diseases. A growing number of IMDs are equipped with a wireless interface that allows non-invasive monitoring and reprogramming through an external device, also known as device programmer. However, this wireless interface also brings important security and privacy risks that may lead to remote attacks. In this domain, the use of cryptography is challenging due to the inherent tensions between security vs. accessibility and security vs. energy cost. A well-studied problem yet unsolved is how to establish (and manage) cryptographic keys between the device programmer and the IMD. Recent work has investigated how Physiological Signals (PS) extracted from the patient can be used for key agreement or authentication between the devices.

This paper surveys some of the proposed countermeasures in the field of medical device security, with a special focus on those that use patient's physiological signals for key establishment or authentication between the devices. We point out that most of the existing solutions, including those relying on PS, take assumptions that do not necessarily hold in practical scenarios. Furthermore, we show that the H2H protocol and the Biosec protocol have serious security weaknesses and design flaws which make them vulnerable to attacks. Based on our analysis, we define some of the challenges that need be addressed before adopting these solutions. Furthermore, we investigate how to use physiological-signal-based protocols in cryptography, possibly in combination with other solutions, such as pre-installed factory keys, to achieve higher security protection.

CCS Concepts: •**Security and privacy** → **Access control; Multi-factor authentication; Security protocols;**

Additional Key Words and Phrases: IMDs, physiological signals, key generation and key agreement, security with noisy data, fuzzy cryptographic primitives

1. INTRODUCTION

Implantable Medical Devices (IMDs) such as pacemakers, Implantable Cardioverter Defibrillators (ICDs) or neurostimulators, help doctors to monitor and treat chronic diseases like arrhythmia or Parkinson. IMDs are resource-constrained devices with a limited battery life that require reduced size, low peak power and a low duty cycle. When the battery is drained, the patient needs to undergo surgery for IMD replacement. The goal of IMDs is not only to improve the quality and life expectancy of patients but also to contribute to the sustainability of health-care systems. For this purpose, the newest IMD contain a radio transceiver that enables wireless communication with external devices. Through device programmers, doctors can wirelessly retrieve patient- and telemetry data or reprogram the IMD without needing to perform invasive surgery on the patients. While convenient, the wireless interface also poses some security and privacy risks that can result in attacks which may compromise the privacy or cause medical injuries to the patient.

Several articles have shown that IMD manufacturers often rely on keeping the protocol specifications secret in order to provide security. This is commonly known as security-through-obscurity, and it is typically used to conceal insecure designs. Halperin et al. examined the proprietary protocol between the device programmer and an ICD to communicate over a short-range communication channel (less than 10 cm) [Halperin et al. 2008b]. As no security mechanisms were found, they were able to perform several software radio-based attacks just by replaying past transmissions sent by legitimate device programmers. Marin et al. analysed the proprietary protocol between the device programmer and the newest model of an ICD over a long-range com-

munication (from two to five meters) [Marin et al. 016b]. Their work revealed serious protocol and implementation weaknesses on ICDs. Moreover, they demonstrated that it is possible to fully reverse-engineer the protocol by using a black-box approach without of physical access to the devices. Li et al. conducted remote attacks on an insulin pump [Chunxiao et al. 2011]. Marin et al. extended the attacks of Li et al. [Chunxiao et al. 2011] and fully reverse-engineered the proprietary protocol between the insulin pump and all its peripherals [Marin et al. 016a].

Paper outline: The remainder of this paper is organised as follows. Section 2 describes the main challenges to secure the communication between the device programmer and the IMD, including the well-known tensions between security vs. accessibility and security vs. energy cost. Section 3 gives an overview of the existing countermeasures. Section 4 reviews the proposed solutions based on the use of physiological signals for key agreement or authentication, whereas novel attacks on the H2H and the Biosec protocol are shown in Section 5. Section 6 provides a framework with a set of recommendations on how to use physiological signals in cryptography, discusses the unrealistic assumptions and threat models that are typically considered, and defines new future research directions. Section 7 gives concluding remarks.

2. HOW TO BALANCE SECURITY, PRIVACY, SAFETY AND UTILITY?

To prevent the attacks discussed above, strong security mechanisms are required between the device programmer and the IMD. However, Halperin et al. identified some inherent tensions between security vs. accessibility and security vs. energy cost which led them to conclude that it is necessary to find a trade-off between security, privacy, safety and utility [Halperin et al. 2008a].

Both the cryptographic protocols and the cryptographic primitives should be as efficient as possible in order to minimise the energy consumption. Otherwise, the security mechanisms may reduce the battery life, reliability and performance of the IMD. The latter three requirements should be considered a top priority when implementing security measures on IMDs. While public-key cryptography can be expensive in terms of both computational time and energy consumption, symmetric cryptography consumes less energy and hence it is suitable for resource-constrained devices such as IMDs. However, symmetric cryptography requires the devices to run a key agreement or a key transport protocol in order to share a secret cryptographic key. This key is then used as parameter in the cryptographic algorithms. Yet, establishing a key between the device programmer and the IMD is a widely studied research problem that still remains unsolved. This is due to the IMD constraints and the challenge to manage the cryptographic keys in terms of scalability, usability and the capacity to deal with emergency situations. For example, consider a cardiac patient who is travelling. While strong security mechanisms are necessary to prevent adversaries from performing attacks, these security mechanisms should still allow doctors to identify and access the IMD in order to treat the patient in an emergency situation.

A naive approach to perform authentication in an emergency situation would be to ask patients to remember a key or a password. This key could then be used to grant permission to doctors to access the patient's IMD in an emergency situation. In emergencies, though, if the patient is unconscious, he will be unable to tell the password to the medical staff. To overcome this problem, the key could be printed on a bracelet worn by the patient or tattooed in his skin. The downside of this solution is that bracelets can be lost, stolen or damaged, and implicitly reveal the patient condition to others, whereas tattoos can become unreadable after an accident. Also, patients may have cultural, social, or personal objections against tattoos, as shown by Denning et al. [Denning et al. 2010]. Another possibility would be to store a master key on all device programmers in tamper-resistant hardware, and have diversified keys pre-

installed in every IMD [Halperin et al. 2008b]. But having the master key stored in every device programmer poses prominent risks. If a single device programmer would ever be compromised, every patient with an IMD will be exposed to attacks until his IMD is replaced. An alternative to the previous solution would be to store this master key securely in the cloud. This way, the device programmer can obtain the key only after performing an authentication process in which doctors may need to prove their identity as well. After finishing each session, the device programmer can erase the key from its memory not to leave any trace to the adversaries. However, this would require the device programmer to be online while the doctor accesses the IMD, which may be unrealistic in some cases.

Unfortunately, key establishment and authentication protocols designed to be employed in traditional networks are not applicable to protect the communication between the device programmer and the IMD. There is a need for novel techniques to establish and manage the keys which increase the level of security and privacy of these devices without endangering the patient's safety.

3. RELATED WORK

In the literature, the countermeasures can be grouped into five main categories: (i) using external devices as shields, (ii) using anomaly detection, (iii) using auxiliary channels, (iv) proximity-based solutions and (v) using physiological signals extracted from the patient for key establishment or authentication. In this section, we will give an overview of the proposed solutions for each of the first four groups. The solutions based on the use of physiological signals will be explained more in detail in the next section. Although we will explain each of these solutions separately, it is possible to combine some of them to achieve a higher security level.

Gollakota et al. proposed to use an external device – which they call "shield"– for protecting legacy IMDs (i.e. those which are already implanted in the patient's body) [Gollakota et al. 2011]. The shield mediates between the device programmer and the IMD and implements a jam-cum receiver that allows to jam the messages to/from the IMD to prevent unauthorised entities from decoding them, while still being able to successfully receive and decode them itself. Although the shield alleviates some security problems, its main disadvantage is that it can disrupt ongoing communications between legitimate devices. In addition, the shield does not protect against high-powered adversaries, and provides weak confidentiality guarantees since a MIMO adversary can cancel out the jamming signal and recover the message content, as shown by Tippenhauer et al. [Tippenhauer et al. 2013]. Xu et al. introduced a wearable device called "IMDGuard" whose function is to authenticate the device programmer on the ICD's behalf through an ECG-based key establishment [Xu et al. 2011]. However, the IMDGuard is vulnerable to a Man-In-The-Middle (MITM) attack which reduces its effective key length from 129 bits to 86 bits, as shown by Rostami et al. [Rostami et al. 2013a]. A 86-bit key is not sufficient for long term security. Zhang et al. proposed "Medmon", a multi-layered anomaly detection system that relies on physical and behavioral anomaly detection to detect malicious actions [Zhang et al. 2013]. Depending on the threat, Medmon can either warn the patient or jam the wireless channel to block the malicious messages. Yet, Medmon is neither flexible nor user-friendly, energy-consuming, and it only offers a weak security protection.

Another type of solutions relies on exchanging a cryptographic key through an auxiliary or Out-Of-Band (OOB) channel. Halperin et al. introduced a zero-power authentication protocol through which the IMD generates and sends a key to the device programmer over an audio channel [Halperin et al. 2008b]. Although the use of a zero-power authentication protocol is promising to prevent some Denial-of-Service (DoS) attacks, Halevi et al. showed that it is possible to eavesdrop the audio channel from

several meters away and successfully recover the key [Halevi and Saxena 2010]. Kim et al. presented a vibration-based key distribution where the device programmer generates and sends a key to the IMD over a vibration channel [Kim et al. 2015]. Furthermore, they proposed a technique to mask the acoustic emanations that are produced as a side effect of the vibration. It remains unclear, though, what the effects are of this masking technique on the key sent over the vibration channel and whether it is possible to eavesdrop the vibration channel. Rasmussen et al. presented an access control system based on ultrasonic distance bounding in combination with the Diffie-Hellman protocol [Rasmussen et al. 2009]. In their solution, the IMD establishes a key with any device programmer that is in its close proximity, meaning that it provides security only under the assumption that the adversary cannot be sufficiently close to the patient. Body-Coupled Communication (BCC) uses the patient's body as a communication medium to transport a key between devices. However, BCC-based solutions are shown to be vulnerable to remote eavesdropping by adversaries with a very sensitive antenna, see e.g. [Chunxiao et al. 2011].

4. SOLUTIONS BASED ON PATIENT PHYSIOLOGICAL SIGNALS

The use of Physiological Signals (PS) extracted from the patient for generating and establishing a secret key between two devices was first proposed by Poon et al. [Poon et al. 2006]. Unlike biometrics, which are (to some extend) time-invariant, PS are random signals that vary over time. Most common PSs for key generation include those originating from the patient's heart such as the ElectroCardioGram (ECG). However, other PSs such as the PhotoPlethysmoGram (PPG), blood glucose, blood pressure, temperature, hemoglobin and blood flow have been proposed as well [Yao et al. 2011]. Poon et al. showed that the time between patient heartbeats, also known as InterPulse Interval (IPI), exhibits two desirable properties: (i) it provides a high level of randomness and (ii) it can be measured anywhere on the body by touching the patient's skin. A common approach for PS-based key agreement protocols is that the device programmer and the IMD take a measurement of the chosen PS, and then use it to generate a shared cryptographic key.

Traditionally, cryptographic keys need to be uniformly distributed and identical on both sides of the communication, as having a slightly different key would not allow the device to decrypt the message correctly. However, the PS measurements taken by the devices are typically not equal but at best only rather similar due to the noise, and not necessarily uniformly distributed. Dodis et al. studied the problem of how to generate cryptographic keys from noisy data [Dodis et al. 2004]. They provided formal definitions and proposed two primitives for securely converting noisy data into cryptographic keys: the *fuzzy extractor* and the *secure sketch*. *Fuzzy extractors* allow to extract nearly uniform randomness R from an input w while tolerating some errors. In other words, given a noisy input value w' it is possible to recover R if w' is rather similar to w. Fuzzy extractors consists of two functions: *generate* and *reproduce*. The former is executed with w as an input, and outputs a key k and helper data P, whereas the latter is executed with w' and P as inputs, and outputs k. *Secure sketches* are similar to fuzzy extractors in the sense that they also tolerate some errors in the cryptographic keys. However, they do not address the non-uniformity distribution of the generated keys.

Juels et al. were the first to propose two practical realisations of these fuzzy cryptographic primitives: the *fuzzy commitment* [Juels and Wattenberg 1999] and the *fuzzy vault* [Juels and Sudan 2006]. They are both based on the use of cryptography in combination with Error Correction Codes (ECC).

Fuzzy commitment scheme: Like conventional commitment schemes, fuzzy commitment schemes are both concealing and binding. The former indicates that adver-

saries cannot open the commitment in order to recover the committed value, whereas the latter refers to the fact that it is not possible to change the value after committing to it. To understand how the fuzzy commitment works, let us give an example. Suppose that a device $D1$ wants to securely transport a key k to device $D2$ using a fuzzy commitment scheme F. We denote the witnesses used by $D1$ and $D2$ as w and w' , respectively. In this context, w and w' are two noisy measurements of the chosen PS taken by each of the devices. Initially, $D1$ generates a random key k and then adds some redundancy to k depending on the type of ECC. This process converts k into a codeword c. $D1$ then computes $\delta = c \oplus w$ in order to hide c while it is transported from $D1$ to $D2$. Subsequently, $D1$ computes $H(c)$, where H is a one-way function (e.g. a cryptographic hash function). This way, c is concealed using the hash function while δ is left in the clear, as shown in Equation (1).

$$F(c, w) = H(c), c \oplus w \tag{1}$$

Unlike conventional commitment schemes, $D2$ can open the commitment by using any witness w' that is close to w under some suitable metric (e.g. Hamming distance). This leads to $c' = \delta \oplus w'$. If the distance between c' and c is less than the maximum number of errors that the ECC can correct, then $D2$ can successfully recover c from c'. To verify if c is the correct value, $D2$ computes $H(c)$ and compares it with the hash value of the commitment sent by $D1$.

Fuzzy vault scheme: It can be seen as an order-invariant version of the fuzzy commitment scheme proposed by Juels et al. The fuzzy vault is designed to lock a secret k into a vault using a set of features A (instead of only one). The vault can then be unlocked only by using a set of features B similar to those on A. The sets of features A and B can be arbitrarily ordered. Like in the fuzzy commitment scheme, the goal is to transport k from $D1$ to $D2$ without revealing the k to the adversaries by using two distinct (but rather similar) sets of features A and B for concealing k.

The first step for $D1$ is to select a polynomial p in a single variable x and embed k in the coefficients of p. Each of the elements of A are the different x-coordinate values. $D1$ then projects each of the elements of A onto points that lie on the polynomial p. We refer to these points as legitimate points. Subsequently, $D1$ creates random points, also known as chaff points, that do not lie in p. The goal of these chaff points is to conceal the legitimate points. Both the legitimate and chaff points, which form the commitment of k, are then mixed and sent to $D2$. Upon receiving the commitment of k, $D2$ attempts to unlock the vault using B. If B has substantial overlap with A, $D2$ can identify what are the legitimate points and reconstruct p. Reconstructing p allows $D2$ to recover k. The security of this scheme is based on the assumption that the adversaries cannot distinguish between the legitimate and the chaff points.

Several articles have proposed to use fuzzy cryptographic primitives for key establishment or authentication. Miao et al. proposed to modify the original fuzzy vault such that it can be applied in the context of body sensor networks [Miao et al. 2010]. For this, they suggested to use the ECC in a different way in order to minimise a specific type of errors. Cherukuri et al. proposed a fuzzy-commitment-based key distribution protocol, also known as Biosec, wherein PSs extracted from the patient's body are used to securely transport a session key between two (implanted) sensors [Cherukuri et al. 2003]. Their protocol enables sensor re-keying and improves the security of the system over time. Rostami et al. [Rostami et al. 2013b] presented Heart-to-Heart (H2H), a commitment-scheme-based pairing protocol through which the device programmer authenticates to the IMD without needing to share any prior secrets. In the next section we will describe some attacks on both the Biosec and the H2H protocol. K Venkatasubramanian et al. introduced a fuzzy-vault-based key agreement protocol – which

they call PPG based Key Agreement (PKA) – that uses PPG signals to enable sensors to agree upon a symmetric cryptographic key [Venkatasubramanian et al. 2008]. K Venkatasubramanian et al. also proposed a fuzzy-vault-based key agreement protocol, also known as Physiological-Signal-based Key Agreement (PSKA), that allows sensors to securely communicate with each other without requiring any initialisation phase or pre-deployment of keys [Venkatasubramanian et al. 2010]. However, Bagade et al. showed that it is possible to break PSKA in a 30-second handshake with an average probability of 30% [Bagade et al. 2013]. Furthermore, PSKA does not provide high security guarantees due to the limitations on the feature size, and requires a complex algorithm to generate the chaff points. Intuitively, one possible way to increase the security level of PSKA would be to increase the number of chaff points. Yet, increasing the number of chaff points could also result in collisions between the chaff points and the legitimate points generated by the other device. Hu et al. proposed a fuzzy-vault-based key agreement protocol called Ordered-Physiological-Feature-based Key Agreement (OPFKA) which aimed to overcome the PSKA limitations [Hu et al. 2013]. However, Rostami et al. demonstrated that the OPFKA is vulnerable to an attack that exploits the use of the hash function to expand the feature size [Rostami et al. 2013a].

In general, fuzzy commitment schemes have a complex process of obtaining the measurements as they have to generate random binary sequences from the measured PS, whereas fuzzy vault schemes can use the PS measurements directly. However, in the fuzzy vault scheme, the calculation and reconstruction of the polynomial are expensive operations; the process of how to conceal and reveal the key for fuzzy commitment scheme is based only on two operations: a XOR and a hash.

When using fuzzy primitives, there can be two types of errors: False Acceptances (FA) and False Rejections (FR). A FA is produced when an adversary gets a valid authentication by presenting a signal close to the legitimate PS, whereas a FR is produced when a legitimate pair of PS does not produce a valid authentication. Therefore, the False Rejection Rate (FRR) is the proportion between the FRs and the number of legitimate pairing attempts, whereas the False Acceptance Rate (FAR) is the proportion between the FAs and the number of adversarial attempts. Fuzzy commitment schemes typically have a higher FAR than fuzzy vault schemes. This is because the fuzzy vault uses a set of features (instead of only one) to conceal the secret; thus this also comes at the cost of having to transmit more bits. In terms of FRR, they both provide similar performance.

5. CASE STUDIES: SECURITY ATTACKS AND DESIGN FLAWS

In the next sections we will demonstrate that some of the solutions previously discussed consider unrealistic threat models that underestimate the adversary capabilities. In this section, we show that both the Biosec and the H2H protocol have some protocol weaknesses and design flaws that could allow for attacks.

5.1. Attacks on H2H

H2H uses (lightweight) public-key cryptography in combination with a commitment scheme, and implements a novel access-control policy called “touch-to-access”. Touch-to-access ensures access to the IMD by any device programmer that can make physical contact with the patient and measure his heart rate. Figure. 1 provides an overview of the H2H pairing protocol. H2H can be divided into two different phases: (i) a secure-channel setup phase and (ii) an authentication phase. A secure but unauthenticated channel is first created between the IMD and the device programmer via TLS. This channel provides confidentiality, integrity and freshness. The IMD takes the role of a TLS client whereas the device programmer acts as a TLS server. The device programmer presents its certificate to the IMD; however the IMD does not verify it. The

authors state that this is to avoid the burden of a Public-Key Infrastructure (PKI). The TLS session outputs a unique and random number s that does not need to be kept secret, e.g. the hash of the TLS master key and the public key.

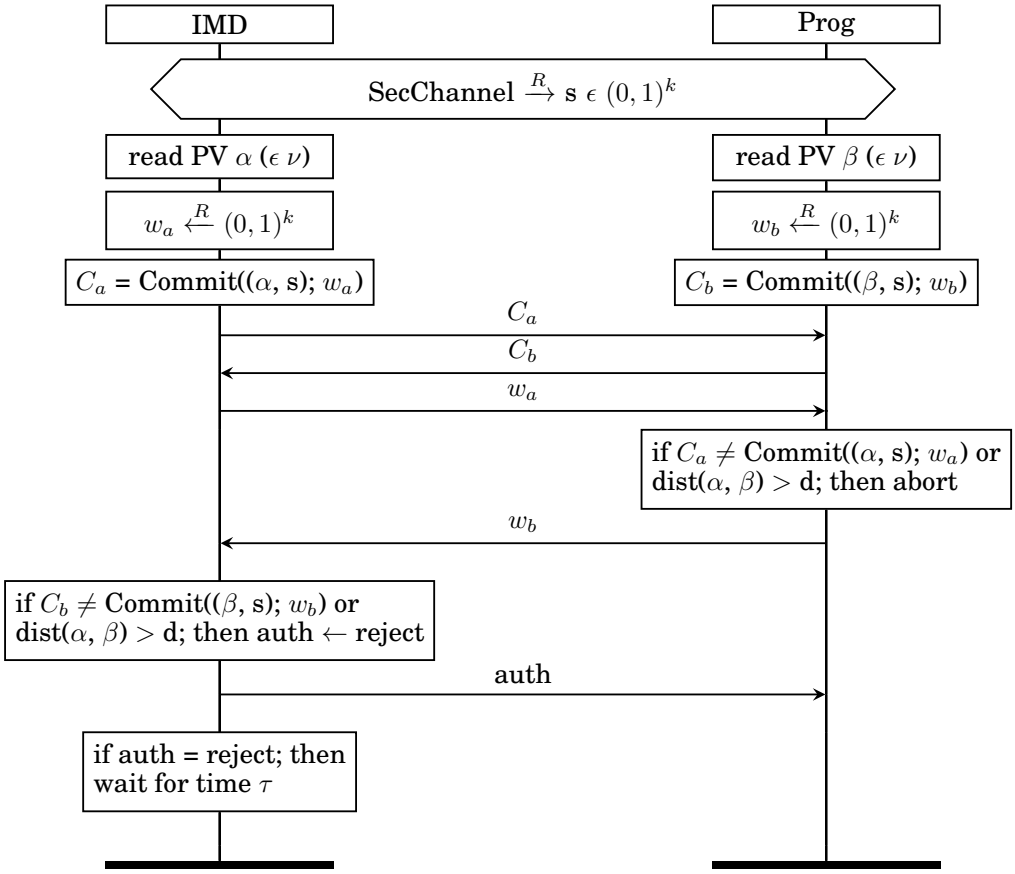

Fig. 1. H2H pairing protocol proposed by Rostami et al.

In the authentication phase, the IMD authenticates the device programmer public key using the touch-to-access policy described above. For this purpose, the IMD and the device programmer each take an IPI reading denoted by α and β, and then generate a random number w_a and w_b respectively. The commitment scheme will output C_a (or C_b) and allows each of the devices to commit to their IPI reading (α or β) while hiding these values by using a random mask w_a (or w_b). Each of the devices binds its commitment also to s in order to avoid re-use of α or β by adversaries on a different channel. Once C_a and C_b have been exchanged, the IMD can open the commitment C_a by sending w_a. This allows the device programmer to obtain α. Subsequently, the device programmer verifies whether α, w_a and s were used to produce C_a, and then checks if the distance between α and β is less than a predetermined threshold. If all these conditions are satisfied, the device programmer sends w_b to open the commitment C_b, which allows the IMD to obtain β. Similarly as before, the IMD checks whether β, w_b and s were used to produce C_b, and checks the distance between α and β. If these conditions are

verified correctly, then the IMD considers the public key of the certificate it received in the secure-channel setup phase as authentic. Unlike passwords, α and β are one-time values and hence can be safely revealed at the end of each protocol instance.

We note that this paper lacks important details about the TLS protocol that is run in the secure-channel setup phase and the commitment scheme that is used in the authentication phase. Without these details, it is difficult to assess the security offered by this protocol. The authors claim that the security of their protocol is based on the assumption that adversaries cannot make physical contact with the patient. They also assume that it is infeasible for adversaries to obtain the patient's IPI by capturing a message sent over the air due to the hiding property offered by the commitment scheme. However, we found two attacks on the H2H protocol that prove that the protocol has serious weaknesses.

Reflection attack: We found a simple yet effective attack where adversaries can gain access to the IMD without needing to know the patient's IPI. Our attack exploits the fact that the H2H protocol is completely symmetric in both directions (i.e. from the IMD to the device programmer and vice versa). In practice, our attack works as follows. The adversary first executes the secure-channel setup phase to set up a TLS session with the IMD. Recall that the IMD does not validate the certificate it receives. In the authentication phase, the goal of the adversary is to authenticate s (the TLS output), which is achieved only by proving knowledge of the patient's IPI. However, we note that adversaries can simply replay the messages sent by the IMD (i.e. choose C_b equal to C_a and w_b equal to w_a). By doing so, the IMD will be convinced that the entity with whom it executes the TLS protocol is a valid device programmer. An easy fix of this flaw is to have the IMD reject C_b in case it is identical to the C_a or to change the protocol such that it is no longer symmetric.

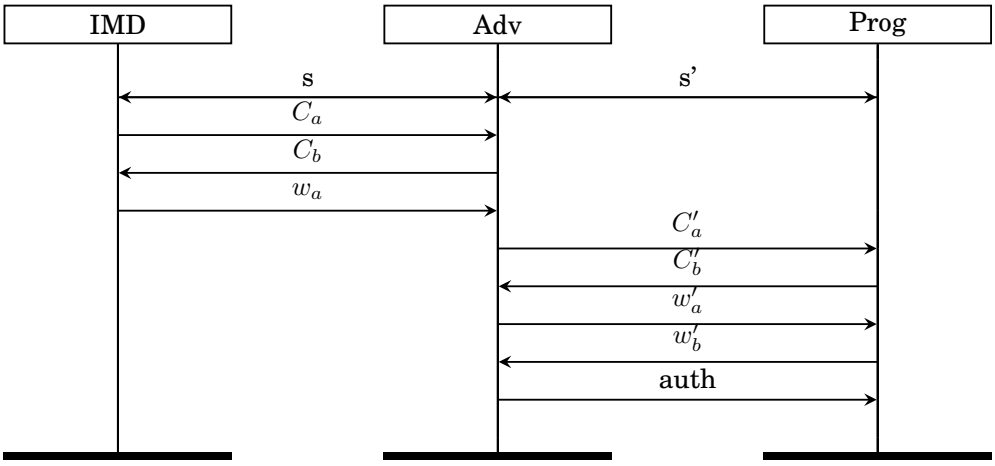

Fig. 2. MITM attack.

MITM attack: The H2H protocol is also vulnerable to a MITM attack in which the adversary makes the device programmer think that it is communicating with the legitimate IMD while it actually does it with a fake IMD (i.e. the adversary). The attack is shown in Figure. 2 and works as follows. The adversary first executes the

secure-channel setup phase with the IMD and the device programmer, respectively, to set up two simultaneous and independent TLS sessions s and s'.

In the authentication phase, the IMD first creates its own commitment C_a and sends it to the device programmer (in that case the adversary). Upon receiving C_a, the adversary creates its own commitment C_b and sends it to the IMD. Both C_a and C_b are sent over s. Since the adversary will abort the session with the IMD before sending the value to open the commitment, C_b can be whatever random value the adversary chooses. The next step for the IMD is to send w_a to allow the opening of the commitment C_a, which enables the adversary to obtain α (the IPI measurement taken by the IMD). As soon as the adversary learns α, he aborts the session with the IMD, creates a new commitment C'_a that contains α, and sends it to the device programmer. The device programmer then creates and sends its commitment C'_b. Both C'_a and C'_b are sent over s'. After receiving C'_b, the adversary sends w'_a, which allows the device programmer to open the commitment C'_a. Following the same procedure, the device programmer sends w'_b such that the IMD (in that case the adversary) can open the commitment C'_b. Finally, the adversary can simply reply with an $auth$ message without needing to open the commitment C'_b. Upon receiving $auth$, the device programmer is convinced that it is communicating with the IMD over s, but actually it is doing it with the adversary over s'.

5.2. Security analysis of Biosec

The goal of the Biosec protocol is to use PSs extracted from the patient's body to securely exchange data between two sensors denoted as $S1$ and $S2$. Fig. 3 provides an overview of the protocol proposed by Cherukuri et al. Before outlining how the protocol works, we first give an overview of the variables and cryptographic primitives used throughout the protocol. $Data$, $eData$ are the data in an unencrypted and encrypted form, respectively, whereas m is a 128-bit MAC tag. K_s is the 128-bit random key generated by $S1$ and transported to $S2$ that is used to perform cryptographic operations. m_s is a 128-bit random number generated by a combination of biometrics whereas r_u is a 128-bit patient's static ID. K_c is the 128-bit number used to mask K_s, and is the result of $m_s \oplus r_u$. $Scom$ is a 128-bit sequence that is the result of masking K_s with K_c. We denote the variables as m'_s, K'_c, K'_s when generated/computed by $S2$. Furthermore we suppose all sensors use an ECC that can correct up to T errors. For this, the following equation needs to be satisfied: T = (D - 1)/2; where D is the minimum distance of the ECC. H is a one-way function (e.g. a cryptographic hash function)

The PS-based key distribution protocol works as follows: $S1$ generates a random session key, K_s, and then encrypts some information, $data$, which results in a ciphertext, $eData$, i.e. $eData = E_{K_s}(Data)$. Subsequently, a 128-bit MAC tag m, is computed over $eData$ ($m = MAC_{K_s}(eData)$). To transport K_s to $S2$ without revealing it to adversaries, $S1$ first adds some redundancy to K_s (depending on the type of ECC), conceals its value using K_c and computes $H(K_s)$. $eData$, m and $Scom$ are then sent to $S2$, where $Scom = \mathrm{H}(K_s) \parallel (K_s \oplus K_c)$. Subsequently, $S2$ attempts to undo the masking operation performed by $S1$ in order to obtain K_s from $Scom$ using K'_c. More specifically, $S2$ reverses the masking operation by computing $K'_s = K_s \oplus K_c \oplus K'_c$. Ideally, if the biometric readings were identical, it would be trivial for $S2$ to obtain K_s. In practice, though, as m'_s is not equal (but rather similar) to m_s, K'_s is slightly different from not than K_s. To recover K_s from K'_s, $S2$ applies the ECC previously used by $S1$. Finally, $S2$ computes $H(K'_s)$ and verifies whether the result corresponds to $H(K_s)$. If this condition is satisfied, $S2$ can use K_s to verify the tag and decrypt the message. The security of their protocol is mainly based on the assumption that adversaries cannot obtain the session key by intercepting a message sent over the air.

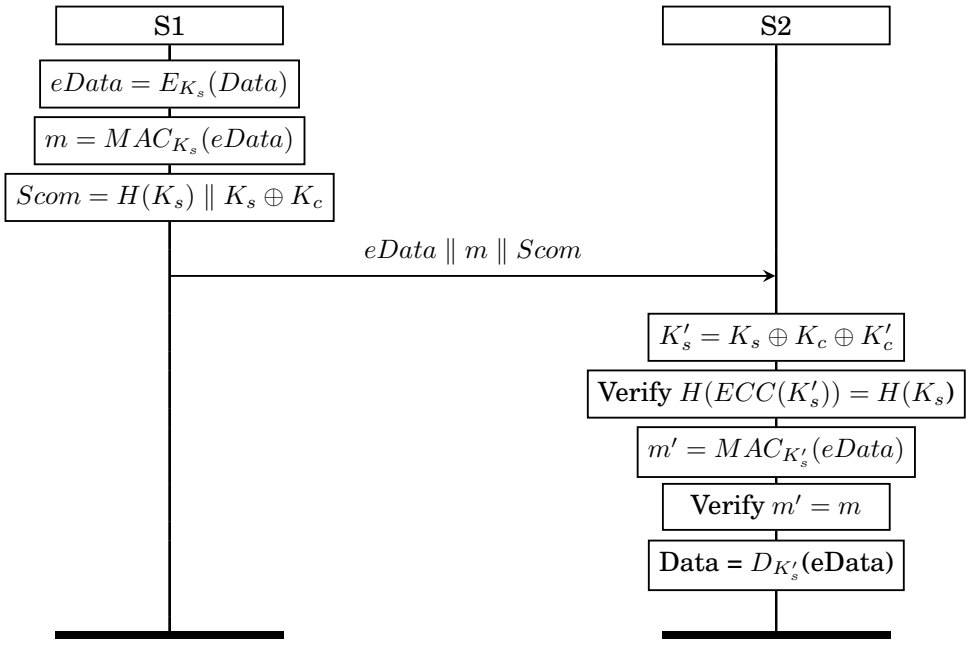

Fig. 3. Key distribution protocol proposed by Cherukuri et al.

We show that the protocol lacks a rigorous security analysis and has important design flaws:

— **K_c randomness**: K_s is XORed with a combination of biometrics and a patient's static ID to prevent adversaries from obtaining this value while it is transported from $S1$ to $S2$. It is unclear what the purpose is of using the static ID and whether/how the receiver sensor knows this ID. However, we note that the static ID does not provide any extra security since the masking operation (i.e. XOR) is linear and hence its effect can be cancelled out just by capturing two messages and subtracting their K_c values.
— **Biometric randomness**: As it has been pointed out by the authors, another possible shortcoming of their solution may be the lack of sufficient entropy.
— **Reuse of the key**: Another problem that we detected is that the authors use only one key for providing confidentiality and authenticity. Using one key in multiple cryptographic primitives is considered to be "bad practice". An easy fix to this problem would be to use K_s as a master key and then derive two independent random keys using a standard key derivation function, or even better, use authenticated encryption.
— **Using an ECC**: The use of an ECC reduces the effective key length as it adds redundancy to correct errors, hence reducing its entropy. A possible solution would be to increase the key length (depending on the ECC) to achieve the same level of security that is expected when no ECC is used. The authors, though, define K_s as a 128-bit key and do not take into consideration this loss of entropy. This may result in brute-force key attacks where adversaries can try all key combinations. This type of attack can be carried out regardless of the randomness level of the biometric and the type of masking operation being used. For each key attempt, adversaries can check whether

they successfully guessed the key as the hash of the key being used is included in the message. In addition, adversaries could potentially perform these computations off-line and create a table with all possible keys and their corresponding hash values. Thus, in practice, adversaries need to capture only one message sent from $S1$ to $S2$, find the match in the table and recover the key.

6. DISCUSSION

6.1. Unrealistic assumptions and threat models

In the previous sections we have discussed which are the current trends and proposed solutions for securing the wireless channel between the device programmer and the IMD. These solutions have tried to address the complex scenarios where the security and privacy of the patients can be compromised, while keeping the utility of the IMD as a priority. They include the use of secured factory-installed keys, the derivation of shared keys from fuzzy PS measurements and proximity-based key exchange protocols. The different proposals are based on different adversarial models, and are mostly designed considering the adversarial limitations, which could be categorised as follows:

— **The adversary cannot extract the key material from a device programmer**, i.e. a secret key can be securely installed in a device programmer [Halperin et al. 2008b], for instance using physically protected storage. This assumption has led to factory-installed secret key approaches.
However, device programmers can be stolen or compromised, and the installed keys could be disclosed by using side-channel attacks (e.g. [Kocher et al. 1999], [Agrawal et al. 2003]). Any factory-installed key material can be hijacked from a stolen device programmer, in a possibly sophisticated but also one-time effort attack. This poses a serious threat to systems which are solely based on factory-installed secrets in the device programmers, as this would permanently compromise the wireless channel.

— **The device programmer will be used only by medical professionals**.
Regarding the threat posed by the compromise of a device programmer, we have mentioned the risk associated to the disclosure of any key material stored in it. However, it should also be considered the risk associated to the use of a device programmer by an unauthorised person. Anyone in possession of a device programmer can use it to reprogram an IMD of an unaware patient. This is indeed the current situation.

— **The adversary cannot be in contact or sufficiently close to the patient during the device programmer-IMD communication**, i.e. proximity between the IMD and the device programmer allows for private and authentic communications. This security based on proximity or contact has led to *proximity-based* secret sharing schemes. Systems such as [Rostami et al. 2013b], [Rasmussen et al. 2009] or [Cherukuri et al. 2003] strongly rely on this proximity-based security assumptions.
Several systems based on the proximity-based secret sharing schemes have been proposed, using electromagnetic or acoustic short-range channels. These systems rely on the assumption that short-range communications cannot be eavesdropped or modified by the adversary. However, proximity communications can be eavesdropped or jammed, as shown in [Halevi and Saxena 2010] for the case of acoustic channels, or [Chunxiao et al. 2011] for the case of electromagnetic channels. These attacks compromise the security of the established channel, demonstrating the risks associated to this unrealistic security assumption.

— **PS cannot be obtained remotely**, i.e. they are accessible only when the device programmer is in direct contact or very close to the patient. The underlying security assumption driving the design of systems using PS as the source of shared secrets is that physiological measurements cannot be obtained unless there exists direct con-

tact or close proximity between the device programmer and the patient. However, several physiological measurements can be accurately acquired from a distance using video cameras, invalidating this assumption. Good examples of these works are [Poh et al. 2011], focused on the estimation of physiological heart rate and respiratory rate, and [Tarassenko et al. 2014], where estimations of heart rate, respiratory rate, and preliminary results on changes in oxygen saturation have been obtained under realistic ambient-light conditions. Furthermore, they demonstrated the feasibility of measuring these physiological variables from a distance. These works show the real weakness of the contact-based assumption for obtaining the physiological measurements, and raise serious doubts on the security of these systems.

— **Resource-constrained devices are capable of performing fuzzy cryptographic primitives**. The use of PSs for cryptographic key binding or generation implies the use of fuzzy cryptographic primitives for dealing with differences in the measurements acquired by the devices. As mentioned in Section 4, practical fuzzy primitives rely on the use of ECCs. The complexity of the decoding procedures depend on the selected code family and parameters, and in some cases can demand too much energy from resource-constrained devices, such as IMDs. For example, Rostami et al. presented in [Rostami et al. 2013b] a system that requires the IMD to perform what they call a decommitment. In practical systems, this procedure involves the use of a decoding algorithm for an ECC, and therefore this approach is relying on the assumption that IMDs are capable to perform this operation. However, the IMD can impose severe limitations in terms of decoding complexity, thus on the error correcting capabilities, and therefore in the effective key length and security of the whole approach.

6.2. New directions

It seems reasonable that the security of the wireless channel between the IMD and the device programmer relies on one of the aforementioned approaches for device programmer authentication or key sharing. However, all of them have intrinsic limitations, as we have discussed. On the one hand, the factory-installed keys framework depends on the security of the key repository (usually stored in the device programmer), which forms a single point of failure. On the other hand, both the use of physiological measurements as source of entropy for a shared secret, and the proximity-based approaches where the channel secrecy is based on the short-range nature of the communication, rely on the assumption that it is difficult for an adversary to get a clear reading (of the physiological measurement or the short-range transmissions) from a distance. As we have discussed, this assumption can also be considered too weak in some cases.

6.2.1. New physiological measurements. In the case of using physiological measurements, more efficient fuzzy primitives and new measurements can be investigated. However, finding a new signal which is suitable for secure secret sharing is not a simple task. In general, this search for new suitable PS must be driven by the following criteria:

— *Entropy*: PSs should provide enough entropy to secure the communication in short periods of time.
— *Readiness*: acquisition of the signals should be a fast process.
— *Distinctiveness*: signals must contain person-specific and session-specific information, making it impossible to use a signal acquired from a different individual or from the same individual but at a different session.

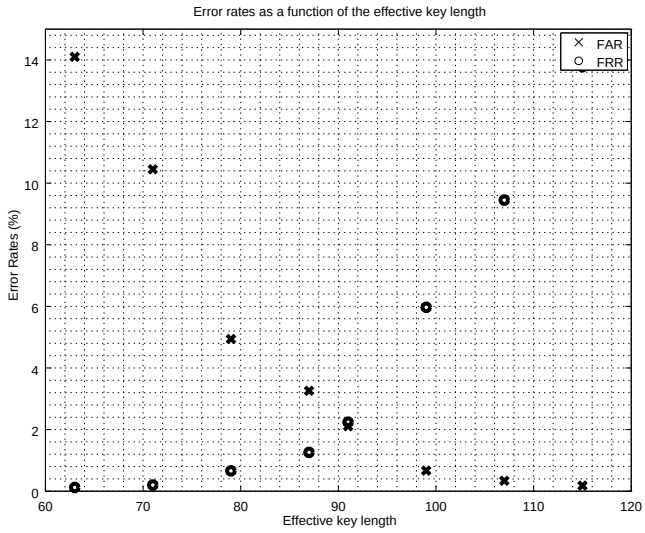

Fig. 4. Example of False Acceptance Rate and False Rejection Rate as a function of the effective key length for 255 code length BCH codes.

— *Precision*: differences between the signals sensed at the IMD and at the device programmer should be as small as possible, thus maximising the effective key length and minimising the entropy loss and the capture time.
— *Exclusivity*: PSs used for device programmer-IMD pairing must be difficult to estimate using remote sensing, but must be readable by both the IMD and the device programmer. Remote sensing attacks must be considered when evaluating the security of these schemes, since at least it could lead to a reduction of the security parameter of the pairing protocol.
— *Availability*: the properties of the PSs should be independent of the patient and its particular condition.

6.2.2. Efficient and flexible fuzzy primitives. The use of fuzzy cryptographic primitives entails an extra energy consumption in both the IMD and the device programmer. This energy consumption can be divided into two main components: the communication and the computation cost. The former indicates the cost of transmitting/receiving bits to/from a device while the latter refers to the cost of performing operations (e.g. the hash when using the fuzzy commitment scheme). Marin et al. [Marin et al. 016a] showed that communication cost is dominant when dealing with low-complexity cryptographic primitives, such as symmetric encryption/decryption and MAC generation/verification. However, the situation can drastically change when dealing with complex operations, such as memory-efficient ECC decoding. Therefore, the energy consumption by the IMD when performing the ECC-related operations must be considered when designing the key-sharing protocol, carefully selecting which primitives are executed by which party and which codes are used.

Another important aspect is how to set the working point of the fuzzy primitives. Figure 4 shows the error rates as functions of the effective key length for a 255-bit synthetic signal generated by assuming constant error rates per bit for the legitimate and the adversarial signals and using BCH ECCs [Hocquenghem 1959], [Bose and Ray-Chaudhuri 1960]. The effective key length is the length of the message encoded by

the ECC in a fuzzy commitment scheme. For example, when the effective key length is 91 and the code length is 255, the code used is a BCH with length 255, dimension 91, and error correcting capability 25. It is easy to see that more error correcting capabilities are required for low FRRs, thus resulting in lower effective key lengths. FRRs are closely related with the availability, since a high FR would make more difficult and time consuming the device pairing.

From this synthetic example, it is easy to see that the fuzzy primitives provide a variety of available working points and security trade-offs for the same signal. As not all the functions and sessions of a device programmer-IMD might have the same requirements, different working points can be set for sessions with different purposes. Session-specific security and availability requirements can be specified in order to select the most appropriate ECC and working point.

6.2.3. Doctor authentication. As previously mentioned, a compromised device programmer could be used by any person, opening an opportunity for attacks. A possible solution would be to incorporate an authentication mechanism for the medical professionals allowed to use the device programmers. For instance, each professional could be provided with an authentication token or smart card with cryptographic capabilities containing a public-private key pair signed by the manufacturer. Unlike IMDs, device programmers could easily check the authenticity of the professional's card using public cryptography primitives, and this would impede unauthorised users from using a stolen device programmer. This solution seems only applicable to the professional authentication for using the device programmer, while it does not help to diminish the risk associated with the disclosure of key material stored in the device programmers, which would still pose a threat to systems where a factory-installed master key is stored in the device programmers.

6.2.4. Hybrid approaches. Regarding the secure establishment of a channel between the device programmer and the IMD, one possible solution to overcome the weaknesses from each approach is to combine them in an hybrid approach, using both factory-installed keys and a proximity or contact-based secret sharing scheme (either short-range communications or physiological measurements). In such systems, the factory-installed keys would not be a single point of failure, since the adversary would also need to get an accurate enough physiological measurement to get access to the session key. Equivalently, having access to the physiological measurements, or simply eavesdropping the short-range communication would not suffice either, i.e. the adversary would need to simultaneously be in possession of the factory-installed key. Finally, the use of hybrid systems using factory-installed keys, short-range communications and physiological measurements must be also considered. Simultaneously circumventing the three security measures can certainly be much more challenging for the adversary, thus providing increased security. However, more research is needed to investigate how to integrate several of these security solutions without considerably increasing the overall energy cost.

7. CONCLUSIONS

This paper has reviewed some of the existing solutions for protecting the wireless communication between the device programmer and the IMD. A promising line of research proposed to use physiological signals (PS) extracted from patients for key establishment or authentication between the devices. However, some of these solutions have shown to have security flaws that make them vulnerable to attacks. Our work revealed serious security weaknesses in the H2H and the Biosec protocol. Another common problem of most of the existing solutions, including those relying on PSs, is that they typically take unrealistic assumptions that underestimate the adversaries capa-

bilities. Furthermore, it still remains unclear if the conventional fuzzy cryptographic primitives can be used in resource-constrained devices like IMDs. Future research should focus on optimising or designing more efficient fuzzy cryptographic primitives. For this purpose, we provided a framework with a set of recommendations on how use physiological signals in cryptography. The observations and lessons learned from this work can facilitate the process of how to design both more efficient fuzzy cryptographic primitives and secure protocols.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by the Research Council KU Leuven: C16/15/058.

REFERENCES

Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. 2003. The EM Side-Channel(s). In *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '02)*. Springer-Verlag, London, UK, UK, 29–45. http://dl.acm.org/citation.cfm?id=648255.752713

P. Bagade, A. Banerjee, J. Milazzo, and S. K. S. Gupta. 2013. Protect your BSN: No Handshakes, just Namaste!. In *2013 IEEE International Conference on Body Sensor Networks*. 1–6. DOI:http://dx.doi.org/10.1109/BSN.2013.6575511

R.C. Bose and D.K. Ray-Chaudhuri. 1960. On a class of error correcting binary group codes. *Information and Control* 3, 1 (1960), 68 – 79. DOI:http://dx.doi.org/10.1016/S0019-9958(60)90287-4

S. Cherukuri, K. K. Venkatasubramanian, and S. K S Gupta. 2003. *Biosec: A biometric based approach for securing communication in wireless networks of biosensors implanted in the human body*. Vol. 2003-January. Institute of Electrical and Electronics Engineers Inc., 432–439. DOI:http://dx.doi.org/10.1109/ICPPW.2003.1240399

Li Chunxiao, A. Raghunathan, and N.K. Jha. 2011. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In *e-Health Networking Applications and Services, 13th IEEE International Conference on*. 150–156. DOI:http://dx.doi.org/10.1109/HEALTH.2011.6026732

Tamara Denning, Alan Borning, Batya Friedman, Brian T. Gill, Tadayoshi Kohno, and William H. Maisel. 2010. 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 (CHI '10)*. ACM, New York, NY, USA, 917–926. DOI:http://dx.doi.org/10.1145/1753326.1753462

Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. 2004. *Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data*. Springer Berlin Heidelberg, Berlin, Heidelberg, 523–540. DOI:http://dx.doi.org/10.1007/978-3-540-24676-3_31

Shyamnath Gollakota, Haitham Hassanieh, Benjamin Ransford, Dina Katabi, and Kevin Fu. 2011. They Can Hear Your Heartbeats: Non-invasive Security for Implantable Medical Devices. *SIGCOMM Comput. Commun. Rev.* 41, 4 (Aug. 2011), 2–13. DOI:http://dx.doi.org/10.1145/2043164.2018438

Tzipora Halevi and Nitesh Saxena. 2010. On pairing constrained wireless devices based on secrecy of auxiliary channels: the case of acoustic eavesdropping. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*. 97–108. DOI:http://dx.doi.org/10.1145/1866307.1866319

Daniel Halperin, Thomas S. Heydt-Benjamin, Kevin Fu, Tadayoshi Kohno, and William H. Maisel. 2008a. Security and Privacy for Implantable Medical Devices. *IEEE Pervasive Computing, Special Issue on Implantable Electronics* 7, 1 (Jan. 2008), 30–39. DOI:http://dx.doi.org/10.1109/MPRV.2008.16

Daniel Halperin, Thomas S. Heydt-Benjamin, Benjamin Ransford, Shane S. Clark, Benessa Defend, Will Morgan, Kevin Fu, Tadayoshi Kohno, and William H. Maisel. 2008b. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proceedings of the 29th Annual IEEE Symposium on Security and Privacy*. 129–142. http://www.secure-medicine.org/icd-study/icd-study.pdf

A. Hocquenghem. 1959. Codes Correcteurs d'Erreurs. *Chiffres (Paris)* 2 (Sept. 1959), 147–156.

C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen. 2013. OPFKA: Secure and efficient Ordered-Physiological-Feature-based key agreement for wireless Body Area Networks. In *INFOCOM, 2013 Proceedings IEEE*. 2274–2282. DOI:http://dx.doi.org/10.1109/INFCOM.2013.6567031

Ari Juels and Madhu Sudan. 2006. A Fuzzy Vault Scheme. *Des. Codes Cryptography* 38, 2 (Feb. 2006), 237–257. DOI:http://dx.doi.org/10.1007/s10623-005-6343-z

Ari Juels and Martin Wattenberg. 1999. A Fuzzy Commitment Scheme. In *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS '99)*. ACM, New York, NY, USA, 28–36. DOI:http://dx.doi.org/10.1145/319709.319714

Younghyun Kim, Woo Suk Lee, Vijay Raghunathan, Niraj K. Jha, and Anand Raghunathan. 2015. Vibration-based Secure Side Channel for Medical Devices. In *Proceedings of the 52Nd Annual Design Automation Conference (DAC '15)*. ACM, New York, NY, USA, Article 32, 6 pages. DOI:http://dx.doi.org/10.1145/2744769.2744928

Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential Power Analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '99)*. Springer-Verlag, London, UK, UK, 388–397. http://dl.acm.org/citation.cfm?id=646764.703989

Eduard Marin, Dave Singelée, Flavio Garcia, Tom Clothia, Rik Willems, and Bart Preneel. 2016(b). On the (in)security of the Latest Generation Implantable Cardiac Defibrillators and How to Secure Them *(Forthcoming)*.

Eduard Marin, Dave Singelée, Bohan Yang, Ingrid Verbauwhede, and Bart Preneel. 2016(a). On the Feasibility of Cryptography for a Wireless Insulin Pump System. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy (CODASPY '16)*. ACM, New York, NY, USA, 113–120. DOI:http://dx.doi.org/10.1145/2857705.2857746

Fen Miao, Shu-Di Bao, and Ye Li. 2010. A Modified Fuzzy Vault Scheme for Biometrics-Based Body Sensor Networks Security. In *Proceedings of the Global Communications Conference, 2010. GLOBECOM 2010, 6-10 December 2010, Miami, Florida, USA*. 1–5. DOI:http://dx.doi.org/10.1109/GLOCOM.2010.5683998

M. Z. Poh, D. J. McDuff, and R. W. Picard. 2011. Advancements in Noncontact, Multiparameter Physiological Measurements Using a Webcam. *IEEE Transactions on Biomedical Engineering* 58, 1 (Jan 2011), 7–11. DOI:http://dx.doi.org/10.1109/TBME.2010.2086456

C. C. Y. Poon, Yuan-Ting Zhang, and Shu-Di Bao. 2006. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Communications Magazine* 44, 4 (April 2006), 73–81. DOI:http://dx.doi.org/10.1109/MCOM.2006.1632652

Kasper Bonne Rasmussen, Claude Castelluccia, Thomas S. Heydt-Benjamin, and Srdjan Capkun. 2009. Proximity-based Access Control for Implantable Medical Devices. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*. ACM, New York, NY, USA, 410–419. DOI:http://dx.doi.org/10.1145/1653662.1653712

Masoud Rostami, Wayne Burleson, Farinaz Koushanfar, and Ari Juels. 2013a. Balancing security and utility in medical devices?. In *The 50th Annual Design Automation Conference 2013, DAC '13, Austin, TX, USA, May 29 - June 07, 2013*. 13:1–13:6. DOI:http://dx.doi.org/10.1145/2463209.2488750

Masoud Rostami, Ari Juels, and Farinaz Koushanfar. 2013b. Heart-to-heart (H2H): authentication for implanted medical devices. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*. 1099–1112. DOI:http://dx.doi.org/10.1145/2508859.2516658

L Tarassenko, M Villarroel, A Guazzi, J Jorge, D A Clifton, and C Pugh. 2014. Non-contact video-based vital sign monitoring using ambient light and auto-regressive models. *Physiological Measurement* 35, 5 (2014), 807. http://stacks.iop.org/0967-3334/35/i=5/a=807

N. O. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun. 2013. On Limitations of Friendly Jamming for Confidentiality. In *Security and Privacy (SP), 2013 IEEE Symposium on*. 160–173. DOI:http://dx.doi.org/10.1109/SP.2013.21

Krishna K Venkatasubramanian, Ayan Banerjee, and S Gupta. 2008. Plethysmogram-based secure inter-sensor communication in body area networks. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*. IEEE, 1–7.

Krishna K. Venkatasubramanian, Ayan Banerjee, and Sandeep Kumar S. Gupta. 2010. PSKA: Usable and Secure Key Agreement Scheme for Body Area Networks. *Trans. Info. Tech. Biomed.* 14, 1 (Jan. 2010), 60–68. DOI:http://dx.doi.org/10.1109/TITB.2009.2037617

Fengyuan Xu, Zhengrui Qin, Chiu Chiang Tan, Baosheng Wang, and Qun Li. 2011. IMDGuard: Securing implantable medical devices with the external wearable guardian. In *INFOCOM 2011. 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 10-15 April 2011, Shanghai, China*. 1862–1870. DOI:http://dx.doi.org/10.1109/INFCOM.2011.5934987

Lin Yao, Bing Liu, Guowei Wu, Kai Yao, and Jia Wang. 2011. A Biometric Key Establishment Protocol for Body Area Networks. *IJDSN* 2011 (2011).

Meng Zhang, Anand Raghunathan, and Niraj K. Jha. 2013. MedMon: Securing Medical Devices Through Wireless Monitoring and Anomaly Detection. *IEEE Trans. Biomed. Circuits and Systems* 7, 6 (2013), 871–881. DOI:http://dx.doi.org/10.1109/TBCAS.2013.2245664