Balancing Security and Utility in Medical Devices?

Masoud Rostami
Rice University
Houston, TX
masoud@rice.edu

Wayne Burleson
University of Massachusetts
Amherst, MA
Burleson@ecs.umass.edu

Ari Juels
RSA Laboratories
Cambridge, MA
ari.juels@rsa.com

Farinaz Koushanfar
Rice University
Houston, TX
farinaz@rice.edu

ABSTRACT

Implantable Medical Devices (IMDs) are being embedded increasingly often in patients' bodies to monitor and help treat medical conditions. To facilitate monitoring and control, IMDs are often equipped with wireless interfaces. While convenient, wireless connectivity raises the risk of malicious access to an IMD that can potentially infringe patients' privacy and even endanger their lives.

Thus, while ease of access to IMDs can be vital for timely medical intervention, too much ease is dangerous. Obvious approaches, such as passwords and certificates, are unworkable at large scale given the lack of central authorities and frequent emergencies in medical settings. Additionally, IMDs are heavily constrained in their power consumption and computational capabilities. Designing access-control mechanisms for IMDs that can meet the many constraints of real-world deployment is an important research challenge.

In this paper, we review proposed approaches to the access-control problem for IMDs, including the problem of secure pairing (and key distribution) between an IMD and another device, such as a programmer. (We also treat related technologies, such as body-area networks.) We describe some limitations of well-conceived proposals and reveal security weaknesses in two proposed cryptographic pairing schemes. Our intention is to stimulate yet more inventive and rigorous research in the intriguing and challenging areas of IMD security and medical-device security in general.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences - Medical information systems; C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based System, Real-time and embedded systems

General Terms

Security, Design, Usability

DAC 13, May 29 - June 07 2013, Austin, TX, USA.

Keywords

Implantable Medical Devices, IMD Security

1. INTRODUCTION

Implantable Medical Devices (IMDs) are increasingly being embedded into patients to monitor medical conditions and to apply a range of therapies, from medication infusion to cardiac pacing and neurostimulation [1]. State-of-the-art IMDs often contain electronic components capable of computation, storage, and wireless communication, and monitor patient conditions in order to adjust their therapeutic regimens. Over the past few decades, IMDs have greatly improved patient care, quality of life, and life expectancy. Next-generation IMDs will provide even more benefit, as they improve existing therapies and enable new ones.

Wireless interfaces contribute significantly to the utility and successful deployment of IMDs, as they enable convenient and non-invasive control, monitoring, and maintenance of the IMD using a control device typically known as a "programmer." A drawback to such wireless access, though, is its inherently open nature, which raises IMDs susceptibility to over-the-air adversarial threats ranging from eavesdropping to unauthorized access and control. IMD manufacturers are subject to strict safety and reliability requirements, and the safety of IMDs, including the problem of unexpected failures, has been a subject of ongoing research [2]. IMDs are not subject, however, to similar standards around *logical security* and *access control*. In many cases, the approach has been protection of IMDs through security-by-obscurity: The main barrier to unauthorized access is no more than secret and proprietary design.

Researchers have consequently demonstrated a range of practical attacks, some executed remotely over the air, permitting unauthorized access to IMDs such as cardiac defibrillators and insulin pumps [3, 4]. These attacks enable an adversary to eavesdrop on IMD communications and in some cases emulate a programmer and modify the therapies applied by IMDs, potentially threatening patient privacy and even patients' health and lives. While there are no documented examples of such attacks "in the wild," the need for better secured access control in IMDs (and robust logical security more generally) is urgent and clear.

One major challenge in designing good access-control and other security mechanisms for IMD is their severely constrained resources. IMDs, like other portable electronics, have limited available battery energy. For IMDs, the situation is particularly problematic, as battery replacement usually entails invasive surgery and removal/replacement of the IMD or its components. Additionally, the desire for minimally invasive implantation favors small form-

factors for IMDs, further limiting battery size. Remote power delivery and energy scavenging, although promising, are currently available in only a very limited set of applications.

Three ongoing trends suggest that energy challenges will persist for IMDs. First, the devices are getting increasingly complex and power-hungry due to demand for new, sophisticated therapeutic and monitoring functionality. Power requirements are even outstripping the benefits of Moore's Law and low-power design techniques, as with smart-phones. Second, IMDs are collecting ever more data as new sensors are added to monitor patient health. Transmitting sensor data from an IMD involves wireless communication, which is power intensive. Third, well designed security protocols, including authentication and code verification require the use of cryptography, and cryptographic primitives are notoriously computation- and power-intensive.

A second major challenge in securing IMDs is the tension between the demands of reliable access on the one hand, and protection against access by an adversary or unauthorized entity on the other. In an emergency, medical personnel may need to monitor or reprogram a patient's IMD immediately and thus access it with as little impediment as possible. But an IMD that can be accessed too easily may be vulnerable to eavesdropping on its data transmissions or tampering with its operation.

The conflicting requirements of security, reliability, and usability in IMDs have given rise to an important and vigorous line of research. Halperin et al. [5] first discussed the security and privacy challenges arising from the resource constraints and inflexibility of existing IMD designs, and highlighted fundamental tensions among privacy, security, safety, and utility in IMDs. Increased networking of embedded devices and the emergence of pervasive health care technologies have also motivated closely related security and privacy research for general sensor networks and body-sensor networks (e.g., [6, 7]), healthcare information technology (e.g., [8]), and patient health data (e.g., [9]).

In this paper, we briefly survey the problem of enabling authorized access to IMDs by programmers. We use the terms IMD and Programmer generically in this paper, but much of the literature we explore treats or is also relevant to other types of medical devices, such as body-area networks (BANs)—arrays of medical / physiological devices that may or many not be implanted.

Security model and goals

The security goals for a device architecture naturally depend upon the participating trustworthy entities and the motives, access, capabilities, and resources of a potential adversary. IMDs typically communicate with a Programmer and potentially with other IMDs or outside-the-body medical devices. The adversary of main concern in these settings is one that acts remotely, over the air, against the IMD's network. Given the open nature of wireless networks, such an adversary may be "active," meaning that it has complete control of the network. Such an adversary can replay, modify, forge, drop, and jam message within the network at will. We can (at least in some cases) assume, however, that one side of the communication, the IMD, is not directly accessible to the adversary, as it is implanted in the body.

Again, a main objective is to allow a Programmer to gain logical access to an IMD while an adversary can't feasibly do so. An obvious approach would be to authenticate an entity communicating with an IMD using a predetermined key or password. The main obstacle to this approach is a fundamental challenge of cryptography: Key distribution. It's impractical to ensure that all valid programmers and/or medical personnel (in medical settings around the world) have valid keys but an adversary can't feasibly gain access to one.

A somewhat more flexible approach to key distribution is the use of public-key cryptography. The TLS protocol, which is ubiquitous on the Internet, relies upon the distribution of public keys to servers. A global public-key infrastructure (PKI) permits designated authorities to certify these public keys, ensuring a binding of the public key to a suitable server identity (domain name). Building a PKI for all medical programmers worldwide—and adequately securing all of their private keys—would be a formidable and probably impractical effort. Recent breakdowns in the trustworthiness of certificate issuance for the Internet, e.g., [10], warn in general of the challenges of constructing sound PKIs.

Recent research on access control and authentication for IMDs and BANs has mainly focused on approaches in which Programmer authorization is determined as a function of *physical access or proximity*. As we explain, there are a number ways to determine whether a Programmer is in suitable proximity to an IMD. The problem often boils down, however, to one of *key-agreement* or *pairing*. Ideally, only an authorized Programmer should be able to establish a (secret) cryptographic key with an IMD; this key enables the establishment of a secure (confidential and integrity-protected) channel between the two devices.

Thus we focus here mainly on proposed approaches to IMD access control through key agreement, which may occur directly between a Programmer and IMD or may be mediated by an additional trusted device carried by a patient. While there are several sound and well-conceived proposed approaches, none in our view provides a fully satisfactory balance of utility and security. Achieving rigorous security guarantees can also be quite challenging: We present attacks against two such proposed protocols, IMD-Guard [11] and OPFKA [12]. Thus, the challenge of good IMD access-control remains an important open research problem.

Organization: The remainder of the paper is organized as follows. We review several IMD key-agreement schemes in Section 2. Distance bounding, jamming, and shielding approaches are discussed in Section 3. In Section 4, we analyze and present attacks against two proposed protocols. We discuss the prospect of using new hardware architectures and device technologies to secure IMDs in Section 5. Section 6 concludes the paper.

2. KEY-AGREEMENT SCHEMES

As we have noted, use of pre-distribution of secret or public keys among IMDs and programmers presents unworkable key-distribution and certification challenges. Proposed access-control protocols for IMDs thus generally avoid reliance on pre-established relationships between IMDs and programmers and instead generate keys on the fly. In this section, we review two general methods proposed in earlier work: (1) Transmitting a secret key using the human body and (2) Key generation using physiological values.

2.1 Key distribution by intra-body signaling

One idea for sharing a secret key between an IMD and Programmer is to generate the key in the IMD and transmit it to the Programmer through the human body itself. This approach requires that the Programmer be in close enough proximity to receive the key; generally, it may make physical contact with a patient. The critical security assumption is that an adversary must operate at a distance from the patient too great to intercept a key; the minimum required distance for such assurance depends on the specific scheme. We now briefly review three proposed intra-body carriers of IMD secret keys: acoustic, electric, and electromagnetic signals.

Acoustic broadcasting. Halperin et al. [3] proposed a scheme in

which an implanted piezo device generates a random key and emits it acoustically. The method results in a rather fast key agreement, requiring only 400ms to transmit a 128-bit key. A serious drawback, however, is the requirement for special implantation of the piezo device. This implantation must be at a depth of 1 cm or so from the skin, ruling out incorporation into deep-body IMDs, such as Implantable Cardioverter Defibrillators (ICDs). The electronic circuits that produce acoustic signals can be shielded with a Faraday cage against electromagnetic interception. An adversary can potentially resort to eavesdropping on acoustic emanations, however, to attack the system. Acoustic eavesdropping of this kind is not well studied and merits further investigation.

Electric and Electromagnetic broadcasting. Zimmerman [13,14] proposed transmitting information through the human body using a pico-amp electric current, in effect using the body as a low-frequency ($\approx$ 1MHz) electrical carrier. Chang et al. [15] discussed securing body area networks (BANs) by distributing a secret key using electrical currents below the action potential of human cells. They used empirical data to analyze the characteristics of the human body as a communication medium. They estimated 0.469 bits per hour as a lower bound on the bandwidth achievable with their proposed method. This is unacceptably slow, of course, for practical IMD key establishment.

In general, key distribution by intra-body communication has the potential to combine strong security against eavesdropping at a distance with minimal power consumption. The actual resistance of such methods to eavesdropping has not been well studied, however. These methods also require approval from government regulators that, to the best of our knowledge, has not yet been granted in the United States even for trial use.

2.2 Key generation using physiological values

The idea of extracting secret keys from physiological values (PVs) to secure IMDs was first suggested in [16]. PVs such as Electrocardiograph (ECG) and Electroencephalography (EEG) signals are suitable candidates for key generation because they provide continuous sources of true randomness. In other words, these PVs may be viewed as entropy sources inside the human body that constantly generate and broadcast (unpredictable) random bits. The randomness of PVs has been documented in an extensive body of medical literature [17–19].

Due to its availability throughout the body and its ease of measurement, the most frequently proposed PV for securing IMDs has been the ECG signal, the electrical signal associated with the activity of the heart.

A number of challenges need to be addressed to achieve practical and secure use of PVs in key agreement. One obstacle is that PV readings are highly sensitive to probe locations on the body and to environmental conditions. Chang et al. [15] assert that if a transformation of the full ECG signal is used for authentication, as suggested in [20, 21], then the PV readings may be so noisy, given a poorly placed probe, that they can be decoded as effectively at a distance by an adversary as by a Programmer with physical contact. The full ECG signal cannot be consistently decoded because the shape of the associated waveform is subject to distortion. Time intervals between specific waveform features, however, can in fact be reliably measured from nearly anywhere in the body. One such feature is the prominent R-Peak of the ECG signal: The time between two R-peaks, which is equal to the heartbeat duration, is essentially invariant to the positioning of probes on the body.

It has been shown that if the heartbeat duration is appropriately quantized, some of its least significant bits are truly random [22–25]. These random bits may differ across probe points on the body due to measurement noise, limiting their naïve use as key bits shared between an IMD and Programmer. To address the challenge of noisy key sources, several methods have been advanced in previous work. For example, Xu et al. [11] proposed the IMDGuard protocol to securely pair an IMD with an external device using noisy ECG data to construct a key. IMDGuard looks to establish a persistent, cryptographic-strength key under non-emergency conditions. Unfortunately, the IMDGuard pairing protocol lacks a rigorous security analysis; Section 4 describes a man-in-the-middle attack that reduces its effective key length and hence its claimed level of security.

Another possible approach for securely extracting a cryptographic key from noisy PV readings is to use "fuzzy" cryptographic primitives, e.g., [26, 27]. Some proposed scheme use the "fuzzy vault" construction in [27] or variants thereof to authenticate devices in a body-area network [20, 21, 28]. Recently, Hu et al. [12] proposed a variant algorithm for PV-based key-agreement, called OPFKA, that is designed to reduce the storage costs associated with fuzzy vaults. OPFKA, however, lacks a rigorous security analysis and, as we explain in Section 4, has notable security weaknesses. The design of a reliable PV-based IMD key-agreement protocol with rigorously analyzed security properties, low power consumption, and a minimal hardware footprint remains a significant open research problem.

3. SECURITY USING DISTANCE BOUNDING OR JAMMING

Another approach to establishing a secure channel between an IMD and a Programmer (or other external device) is to make use of a trusted device to intermediate access to the IMD. This trusted device can be external to a patient's body, and thus well resourced. It can shield the IMD from unauthorized attempts at access and even potentially jam malicious ones.

The idea of blocking inappropriate access to an IMD was first proposed in [29] via a device called a Communication Cloaker. The idea saw a follow-up exploration by Gollakota et al. [30]. Their proposed device, called a *shield*, is worn near the body and used to authenticate / mediate Programmer (or other) communications with the IMD. Helpfully, a shield doesn't require modification of existing IMDs. It protects communications with the IMD using a full duplex radio device acting as a jammer-cum-receiver. It simultaneously listens to and jams IMD messages as appropriate, as well as unauthorized Programmer commands.

Shen et al. [31] have recently proposed a smart jamming technique in which the shield jams the communication channel intermittently; a trusted Programmer knows in advance the intervals in which the channel is clear, and can thus communicate with the IMD. With this solution, the patient has the option of keeping the shield active even during Programmer communication with IMDs.

These approaches have the advantage of being compatible with legacy IMD, so they can be applied seamlessly to the currently deployed devices. A drawback, however, is that jamming, when employed to counteract attacks as in [11, 30], can disrupt the communications of other RF devices and violate laws regarding radio interference.

A promising related approach, by Rassmussen et al. [32], uses ultrasound-based distance bounding to authenticate Programmer access to an IMD, achieving an access policy of proximity similar to those in Section 2 that use intra-body key transmission or key establishment using PVs. Their system requires RF shielding, however, amplifying the engineering complexity of an IMD. It also relies on RF communication. For some IMDs, e.g., brain

implants, RF antennas are of prohibitive length, and alternatives, e.g., infrared, are preferred. Distance-bounding protocols' security models have also historically proven fragile (see, e.g., [33, 34]). Finally, this approach uses ultrasound transmission, which usually requires more power than RF transmission.

4. CASE STUDIES: SECURITY FLAWS

We now describe security weaknesses in two proposed protocols for authenticated key-agreement between devices in a body-area network. One is the setup protocol for the IMDGuard system [11], which pairs a protective device called a "Guardian" with an IMD. The other, OPFKA (Ordered-Physiological-Feature-based Key Agreement), is a generic body-area network pairing protocol [12]. Both IMDGuard and OPFKA rely on ECG measurements as a common source of entropy to establish shared secret keys. While terminology differs across papers, and some involve devices in BANs, we continue to refer to devices generically as the Programmer and IMD.

4.1 Attack on IMDGuard

We briefly describe the IMDGuard scheme for key agreement between a Programmer (in IMDGuard, the Guardian) and IMD. We then show a simple man-in-the-middle attack that reduces the effective key length from 129 bits to 86 bits.

IMDGuard key-agreement protocol The Programmer and IMD each measure ECG data in a succession of four-bit blocks. Let $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ and $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)$ denote respective measurements of one such block by the IMD and Programmer. As these readings are noisy, IMDGuard includes the following noise-reducing "reconciliation" scheme for extraction of key material.

In Round 1, the Programmer and IMD exchange parity bits: The IMD sends $\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4$, while the Programmer sends $\beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4$, where $\oplus$ denotes XOR. If the parity bits agree, the two devices accumulate the first three bits as key material. (They discard a bit to compensate for the one bit leaked by parity-symbol disclosure.) Once 43 blocks pass the parity check, the two sides each possess 129 bits of key material. They hash their respective key material to generate check values h_α and h_β, which they then exchange. The Programmer compares these check values and sends an `accept` message if $h_\alpha = h_\beta$, and a `reject` message otherwise.

Round 2 takes place if (and only if) the Programmer determines that $h_\alpha \neq h_\beta$ (`reject`). In Round 2, the IMD transmits $\alpha_3 \oplus \alpha_4$ and the Programmer, $\beta_3 \oplus \beta_4$. If these two bits agree, the IMD retains α_2 and α_3 as key material, while the Programmer retains β_2 and β_3. I.e., the first bit of each block is discarded to compensate for parity-bit leakage. Further blocks are read and reconciled as needed. When enough bits have accumulated, check values are again compared. (The authors assert a Round 3 is never required.)

Man-in-the middle attack A man-in-the-middle adversary Adv can do the following. Adv allows the IMD and Programmer to proceed normally with parity-bit exchange in Round 1. Suppose that $h_\alpha = h_\beta$ (as happens with high probability). Adv makes two message substitutions at the end of the round: (1) Adv substitutes a random value for the check value h_α transmitted by the IMD, causing the Programmer to send a Round-1 `reject` message and proceed to Round 2 and (2) Adv substitutes an `accept` message for the Programmer's `reject` message, causing the IMD to terminate the protocol with the key established in Round 1.

The Programmer thus proceeds with Round 2. It transmits a second parity bit ($\beta_3 \oplus \beta_4$) for each block from Round 1; at the same time, Adv simulates Round-2 parity-bit transmissions by the IMD. Adv intercepts Programmer parity-bit transmissions to recover an additional bit of information for each block. (For a given block α, the IMD uses $(\alpha_1, \alpha_2, \alpha_3)$ as key bits. Adv learns $\alpha_1 \oplus \alpha_2$.)

While the resulting effective key length of 86 bits is an infeasible target for brute-force attack today, this attack demonstrates a serious design weakness in IMDGuard.

4.2 Attack on OPFKA

In OPFKA, the IMD and Programmer each perform local ECG readings on a human subject over the same interval of time. They translate these readings into a temporally ordered sequence of "features," short (e.g., 12-bit) values. The two devices exploit overlap in their respective feature sequences to construct a shared secret key κ, much as with IMDGuard.

OPFKA adopts a different approach than IMDGuard, however, to specify this overlap. In OPFKA, the Programmer transmits its features obscured with spurious *chaff* values to the IMD in what is called a *coffer*. The IMD indicates to the Programmer those feature values in the coffer that lie in its own feature sequence. Each device, then, can determine the intersection of their two respective feature sequences which is used to construct the shared key κ.

Here is a more detailed specification of the protocol. For simplicity, we assume 12-bit features, one option in OPFKA. We omit protocol parameters and messages not germane to our analysis. For clarity, we also change some of the original notation for OPFKA.

1. **Feature reading:** Each device reads a sequence of N features. (In OPFKA, $N = 30$.) Let $\tilde{F}^{imd} = \{\tilde{f}_0^{imd}, \tilde{f}_1^{imd}, \ldots \tilde{f}_{N-1}^{imd}\}$ be the IMD's features, in temporal order, and $\tilde{F}^{pro} = \{\tilde{f}_0^{pro}, \tilde{f}_1^{pro}, \ldots \tilde{f}_{N-1}^{pro}\}$, the Programmer's.

2. **"Hashing":** Feature values in $\tilde{F}^{imd}$ and $\tilde{F}^{pro}$ are mapped into 20-bit feature values via a "hash" function $H : \{0,1\}^{12} \rightarrow \{0,1\}^{20}$. Let $F^{imd} = \{f_0^{imd}, f_1^{imd}, \ldots f_{N-1}^{imd}\}$ be the resulting set of feature values for the IMD and F^{pro} similarly for the Programmer.

3. **Coffer transmission:** The Programmer randomly selects M chaff features $F' = \{f'_0, f'_1, \ldots, f'_{M-1}\}$, where $f'_i \in_R \{0,1\}^{20} - F^{pro}$. It randomly permutes elements in $C = F \bigcup F'$ and sends the resulting *coffer* C to the IMD.

4. **Coffer opening:** Starting with an empty set J, for each element $f_j^{imd} \in F^{imd}$, the IMD adds j to J if $f_j^{imd} \in C$. The result is an ordered set $J = \{j_0, \ldots, j_{n-1}\}$ of feature positions in F^{imd}. Opening is considered successful if $n \geq q$ for some predetermined parameter q.

5. **Key computation:** The IMD computes $\kappa = h(f_{i_0}^{imd} \parallel f_{i_1}^{imd}, \ldots, \parallel f_{i_n}^{imd})$ for a hash function h. The IMD sends (J, m, μ) to the Programmer, where $\mu = MAC_\kappa[m]$ for a message m (whose details are unimportant here).

Attack on small hash range. OPFKA has a security weakness resulting from the use of hashing in step 2. If the IMD selects in step 4. ("Coffer opening") a feature that is in $C \bigcup F^{imd}$, but not in F^{pro}, then the Programmer cannot then compute κ, and the protocol fails. To reduce the rate of such failures, the authors intend for step 2. ("Hashing") to expand the range of feature values in C.

But application of a "hash" function H *does not* expand the possible range of feature values *for a fixed domain* D. Let $D = \{0,1\}^{12}$ be the set of possible values for a 12-bit feature $\tilde{f}$. The hash of $\tilde{f}$ is computed as $H(s, \tilde{f})$, for pre-agreed salt s (a random

nonce). Let $R = \{H(s, \tilde{f})\}_{\tilde{f} \in D}$ denote the range of $H(s, \cdot)$ over D. Then it is easy to see that $|R| \leq |D| = 2^{12}$, as $H(s, \cdot)$ is a deterministic function. (In fact, given the 12-bit domain and 20-bit range of H in OPFKA, with high probability over s, $|R| < |D|$.)

Thus the vast majority of chaff values in C will be invalid feature values lying outside R. Let $\hat{R} = C \bigcap R$ denote the set of values in the coffer that are valid feature values. (Note that $F_P \subseteq \hat{R}$.) The probability that a randomly selected chaff value $f' \in \{0,1\}^{20} - F^{pro}$ lies in $\hat{R}$ is bounded above by $|R|/(2^{20} - N) < 0.004$.

By excluding invalid chaff values (those not in $\hat{R}$), an adversary can greatly constrain its search space in a brute-force attack against the key κ, as shown in Algorithm 1. (Here, Π_n denotes the set of permutations over $\mathbb{Z}_n$ and $\pi \in \Pi_n$ is a permutation $\pi : \mathbb{Z}_n \leftrightarrow \mathbb{Z}_n$.)

Algorithm 1 Key search algorithm for OPFKA

```
Inputs: J, m, μ, C, R̂
Output: Key κ

for all < f_0, f_1, ..., f_n > ∈ R̂^n do
    for all π ∈ Π_n do
        κ' ← h(f_π(0) ‖ f_π(1) ... ‖ f_π(n−1));
        if MAC_κ'[m] = μ then output κ'; halt
        end if
    end for
end for
```

For example, for one proposed parameterization for OPFKA ($M = 1000$, $N = 30$, and $q = 12$) a key-strength equivalent of 120 bits is claimed in [12]—well beyond feasibility for a brute-force attack. With probability about 63%, though, there will be at most 4 valid chaff values in $\hat{R}$. In this case, assuming $n = q$, the maximum running time of Algorithm 1 will be $\binom{(4+30)}{12} \times 12! \approx 2^{58}$—equivalent to breaking a 58-bit key, and requiring vastly less effort than the claimed 120-bit strength of OPFKA. Cracking a 58-bit key is within the realm of feasibility, as shown by successful cracking of a 64-bit (RC5) key in 2002 [35].

Remark: Distinct salt values might be used in hashing for different positions. This might seem more secure, but isn't, as F^{pro} would no longer in general contain valid feature values for all positions.

Adaptive attack. For large M, such as the proposed parameter $M = 5000$, OPFKA is vulnerable also to an adaptive attack in which an adversary simulates a Programmer to extract the key κ from the IMD. Due to lack of space, our description is brief.

Adv constructs a coffer C as follows. R is partitioned into (arbitrary) equal sized (size 2^{11}) sets R_0 and R_1. A coffer C is constructed that includes R_0 and a subset $R_1' \subseteq R_1$ (to be specified); other features in C are selected to be invalid (drawn from $\{0,1\}^{20} - R$). With high probability, F^{imd} will include $n \geq q$ feature values in R_0. Therefore the IMD will respond to transmission of C with a set of indices J for any choice of R_1'.

Now, in an initial transmission, Adv sets $R_1' = R_1$. With high probability, F^{imd} will include at least one feature value in R_1' with index j. By recursing on halves of R_1' and observing whether $j \in J$ in the IMD's response, Adv can perform a binary search and learn f_j^{imd} with $log_2|D| = 12$ transmissions. By choosing different initial partitions (R_0, R_1), Adv can learn q feature values in F^{imd} and successfully impersonate a legitimate Programmer.

Variant attacks are possible with smaller M and with parallelization to search for multiple IMD feature values simultaneously.

The attack here assumes an ability to query the IMD fairly rapidly, and arises in part because OPFKA includes no throttling or back-off mechanism. A simple countermeasure is for the IMD, after a failed key-agreement with a Programmer, to refuse connections until it collects a fresh set of IPIs. Of course, this raises the risk of denial-of-service attacks and delays due to protocol failures.

5. TOWARD STRONGER AUTHENTICATION TECHNOLOGIES

Perhaps the most significant design constraint on pairing protocols for IMDs are computational power and bandwidth—and thus battery power and energy. With more resources would come a richer design space for signal processing algorithms and also for cryptographic protocols, including various secure two-party computation schemes capable of handling noisy key material, e.g., [36].

Given a specific primitive (e.g., AES) and IMD platform, implementation optimization can save only a fairly limited amount of energy. Thus, enhancing the security capabilities and/or limited battery lifetime of an IMD significantly would require a major change in underlying hardware or energy supply technology. This section discusses three active research areas that promise to extend IMD security capabilities and deployment lifetimes.

Memory technology IMDs typically contain multiple sensors on whose output they perform intensive data processing, storage, and retrieval. This trend, and the energy cost of existing memory devices, argue a need for improved storage technologies. For instance, novel nonvolatile memory technologies with zero leakage power and low read/write energy costs could drastically reduce an IMD's energy consumption; several ongoing research and development efforts, including those in Phase Change Memory (PCM) and Spin-Transfer Torque Random-Access Memory (STT RAM), aim to realize such efficient nonvolatile memory structures [37].

Energy storage technology Active IMDs often obtain power from conventional electro-chemical batteries. Such batteries suffer from slow charge cycles, limited lifetimes, limited power density, and slow rates of improvement in energy capacity. Newer energy-supply technologies, such as nano-scale super-capacitors and fuel cells, with improved cost, size, energy/power storage density, and recycling ability are becoming available [38]. Development, integration, and operation of such novel energy supply devices for IMDs pose an interesting and significant research challenge.

Energy scavenging technology Instead of using an attached energy source internal to the patient's body, IMDs can harvest energy from external sources such as the patient's physical movement, ambient heat, light, radio, or vibrations [39]. There are at least two sets of challenges in the use of energy-scavenging solutions for IMDs. First, the small energy output of typical energy scavenging devices is insufficient for most IMD applications. Second, the inherent uncertainty in harvesting energy from external sources conflicts with IMDs' strict safety and reliability requirements. Nevertheless, development of new energy harvesting, storage, and transfer methods is an active research area which could in time potentially bring significant improvement in energy availability for IMDs.

6. CONCLUSIONS

Presently available IMDs can be wirelessly accessed under loose security policies, allowing attackers to endanger the health and privacy of patients. This paper addressed the problem of authenticating IMDs to external Programmers. Particularly, we reviewed the problem of secure pairing and key distribution between an IMD and Programmers. Several of the currently available proposals to secure IMDs were analyzed. We presented attacks against two such protocols, namely IMDGuard [11] and OPFKA [12]. Securing IMDs

remains to be a challenging open research problem which calls for the development of new and innovative solutions.

7. ACKNOWLEDGMENT

This research was supported in part by an Office of Naval Research grant to the ACES lab at Rice University (ONR R16480).

8. REFERENCES

[1] W. Burleson, S. S. Clark, B. Ransford, and K. Fu, "Design challenges for secure implantable medical devices," in *Proceedings of Design Automation Conference*, pp. 12–17, 2012.

[2] W. Maisel, "Safety issues involving medical devices," *Journal of the American Medical Association*, vol. 294, pp. 955–958, Aug. 2005.

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

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

[5] D. Halperin, T. Kohno, T. Heydt-Benjamin, K. Fu, and W. Maisel, "Security and privacy for implantable medical devices," *IEEE Pervasive Computing*, vol. 7, pp. 30–39, Jan.-Mar. 2008.

[6] F. Stajano and R. Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks," in *Int. Workshop of Security Protocols*, pp. 172–194, 1999.

[7] S. Warren, J. Lebak, J. Yao, J. Creekmore, A. Milenkovic, and E. Jovanov, "Interoperability and security in wireless body area network infrastructures," in *IEEE Engineering in Medicine and Biology Society*, pp. 3837–3840, 2005.

[8] M. Meingast, T. Roosta, and S. Sastry, "Security and privacy issues with health care information technology," in *IEEE Engineering in Medicine and Biology Society*, pp. 5453–5458, 2006.

[9] F. Hu, Q. Hao, M. Lukowiak, Q. Sun, K. Wilhelm, S. Radziszowski, and Y. Wu, "Trustworthy data collection from implantable medical devices via high-speed security implementation based on IEEE 1363," *IEEE Trans. on Info. Tech. in Biomedicine*, vol. 14, pp. 1397–1404, Nov. 2010.

[10] K. Zetter, "DigiNotar files for bankruptcy in wake of devastating hack," *Wired*, 20 Sept. 2011.

[11] F. Xu, Z. Qin, C. Tan, B. Wang, and Q. Li, "IMDGuard: Securing implantable medical devices with the external wearable guardian," in *Proc. of IEEE INFOCOM*, pp. 1862–1870, 2011.

[12] C. Hu, X. Cheng, F. Zhangand, D. Wuand, X. Liao, and D. Chen, "OPFKA: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in *Proc. of IEEE INFOCOM, To Appear*, 2013.

[13] T. G. Zimmerman, "Personal area networks: near-field intrabody communication," *IBM systems Journal*, vol. 35, pp. 609–617, 1996.

[14] T. G. Zimmerman, J. R. Smith, J. A. Paradiso, D. Allport, and N. Gershenfeld, "Applying electric field sensing to human-computer interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 280–287, 1995.

[15] S.-Y. Chang, Y.-C. Hu, H. Anderson, T. Fu, and E. Y. Huang, "Body area network security: robust key establishment using human body channel," in *Proceedings of the USENIX conference on Health Security and Privacy*, pp. 5–5, 2012.

[16] S. Cherukuri, K. Venkatasubramanian, and S. Gupta, "Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in *Parallel Processing Workshop*, pp. 432–439, 2003.

[17] A. L. Goldberger, D. R. Rigney, and B. J. West, "Chaos and fractals in human physiology," *Scientific American*, vol. 262, pp. 42–49, 1990.

[18] M. Signorini, F. Marchetti, and S. Cerutti, "Applying nonlinear noise reduction in the analysis of heart rate variability," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 2, pp. 59–68, 2001.

[19] R. Yulmetyev, P. Hänggi, and F. Gafarov, "Quantification of heart rate variability by discrete nonstationary non-Markov stochastic processes," *Physical Review E*, vol. 65, no. 4, p. 046107, 2002.

[20] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "PSKA: usable and secure key agreement scheme for body area networks," *IEEE Trans. on Information Technology in Biomedicine*, vol. 14, no. 1, pp. 60–68, 2010.

[21] K. K. Venkatasubramanian, A. Banerjee, and S. Gupta, "Plethysmogram-based secure inter-sensor communication in body area networks," in *IEEE Military Communications Conference*, pp. 1–7, 2008.

[22] C. Poon, Y. Zhang, and S. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 73–81, 2006.

[23] S. Bao, C. Poon, Y. Zhang, and L. Shen, "Using the timing information of heartbeats as an entity identifier to secure body sensor network," *IEEE Trans. on Info. Tech. in Biomedicine*, vol. 12, no. 6, pp. 772–779, 2008.

[24] K. Venkatasubramanian and S. Gupta, "Physiological value-based efficient usable security solutions for body sensor networks," *ACM Trans. Sensor Networks*, vol. 6, pp. 31:1–31:36, July 2010.

[25] K. Cho and D. Lee, "Biometric based secure communications without pre-deployed key for biosensor implanted in body sensor networks," in *Information Security Applications*, pp. 203–218, 2012.

[26] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97–139, 2008.

[27] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[28] X. Liao, "Body area network security: A fuzzy attribute-based signcryption scheme," *IEEE Journal on Selected Areas in Communications, To Appear*, 2013.

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

[30] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: non-invasive security for implantable medical devices," in *ACM SIGCOMM*, pp. 2–13, 2011.

[31] W. Shen, P. Ning, X. He, and H. Dai, "Ally friendly jamming: How to jam your enemy and maintain your own wireless connectivity at the same time," in *IEEE Symp. on Security and Privacy*, 2013.

[32] K. Rasmussen, C. Castelluccia, T. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *Proc. of Computer and communications security*, pp. 410–419, 2009.

[33] C. Cremers, K. Rasmussen, B. Schmidt, and S. Capkun, "Distance hijacking attacks on distance bounding protocols," in *IEEE Symp. on Security and Privacy*, pp. 113–127, 2012.

[34] M. Poturalski, M. Flury, P. Papadimitratos, J.-P. Hubaux, and J.-Y. L. Boudec, "Distance bounding with ieee 802.15.4a: Attacks and countermeasures," *IEEE Trans. on Wireless Comms.*, vol. 10, no. 4, pp. 1334– 1344, 2011.

[35] distributed.net, "Project RC5." `http://www.distributed.net/RC5`, Referenced 2012.

[36] B. Kanukurthi and L. Reyzin, "Key agreement from close secrets over unsecured channels," in *Eurocrypt*, pp. 206–223, 2009.

[37] G. Burr, B. Kurdi, J. Scott, C. Lam, K. Gopalakrishnan, and R. Shenoy, "Overview of candidate device technologies for storage-class memory," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 449–464, 2008.

[38] F. Koushanfar and A. Mirhoseini, "Hybrid heterogeneous energy supply networks," in *IEEE Int. Symp. on Circuits and Systems*, 2011.

[39] J. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, 2005.