A System Architecture, Processor, and Communication Protocol for Secure Implants

CHRISTOS STRYDIS, Neuroscience Dept., Erasmus MC & SCT Dept., Delft University of Technology
ROBERT M. SEEPERS, Neuroscience Dept., Erasmus MC
PEDRO PERIS-LOPEZ, CS Dept., Universidad Carlos III de Madrid
DIMITRIOS SISKOS, SCT Dept., Delft University of Technology
IOANNIS SOURDIS, CSE Dept., Chalmers University of Technology

Secure and energy-efficient communication between Implantable Medical Devices (IMDs) and authorized external users is attracting increasing attention these days. However, there currently exists no systematic approach to the problem, while solutions from neighboring fields, such as wireless sensor networks, are not directly transferable due to the peculiarities of the IMD domain. This work describes an original, efficient solution for secure IMD communication. A new implant system architecture is proposed, where security and main-implant functionality are made completely decoupled by running the tasks onto two separate cores. Wireless communication goes through a custom security ASIP, called SISC (Smart-Implant Security Core), which runs an energy-efficient security protocol. The security core is powered by RF-harvested energy until it performs external-reader authentication, providing an elegant defense mechanism against battery Denial-of-Service (DoS) and other, more common attacks. The system has been evaluated based on a realistic case study involving an artificial pancreas implant. When synthesized for a UMC 90nm CMOS ASIC technology, our system architecture achieves defense against unauthorized accesses having *zero energy cost*, running entity authentication through harvesting only $7.45\mu J$ of RF energy from the requesting entity. In all other successfully authenticated accesses, our architecture achieves secure data exchange without affecting the performance of the main IMD functionality, adding less than 1‰ ($1.3mJ$) to the daily energy consumption of a typical implant. Compared to a singe-core, secure reference IMD, which would still be more vulnerable to some types of attacks, our secure system on chip (SoC) achieves high security levels at 56% energy savings and at an area overhead of less than 15%.

Categories and Subject Descriptors: K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Access controls, Authentication*; C.0 [**Computer Systems Organization**]: General—*Instruction set design (e.g., RISC, CISC, VLIW), System architectures*; J.3 [**Life and Medical Sciences**]: Health

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Implantable device, security, ultra-low power, system on chip

This work has been supported by the FP7 European project DeSyRe (grant agreement no: 287611).
Authors' addresses: C. Strydis and R. M. Seepers, Dept. of Neuroscience, Erasmus Medical Center, Dr. Molewaterplein 50, 3015 GE, Rotterdam, The Netherlands; P. Peris-Lopez, Dept. of Computer Science, Universidad Carlos III de Madrid, Av. de la Universidad, 30, 28911 Leganés, Madrid, Spain; D. Siskos, Computer Engineering Lab, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands; I. Sourdis, Dept. of Computer Science and Engineering, Chalmers University of Technology, Rännvägen 6, S-412 96 Gothenburg, Sweden.

DOI: http://dx.doi.org/10.1145/2555289.2555313

ACM Reference Format:
Strydis, C., Seepers, R. M., Peris-Lopez, P., Siskos, D., and Sourdis, I. 2013. A system architecture, processor, and communication protocol for secure implants. ACM Trans. Architec. Code Optim. 10, 4, Article 57 (December 2013), 23 pages.
DOI: http://dx.doi.org/10.1145/2555289.2555313

1. INTRODUCTION

The main purpose of an Implantable Medical Device (IMD) is to measure and/or treat physiological quantities within the body. Millions of people are currently carrying IMDs [Roger et al. 2011], prominent instances of which are *pacemakers* regulating heart rate beating, *IntraCardiac Defibrillators (ICDs)* managing cardiac arrhythmia, *neurostimulators* treating Parkinson's disease, and *cochlear implants* restoring hearing. Modern IMDs are being increasingly equipped with wireless communication capabilities [Gollakota et al. 2011; Halperin et al. 2008] due to introducing certain benefits, primarily as follows [Varshney 2003]:

—*Improved at-home health care:* Patients will not need to visit their doctor as frequently for check-ups; implants will periodically download data wirelessly to a base station at home and, from there, transmit it over the Internet to the doctor's office computer or PDA.
—*Up-to-date device functionality:* Implants will be in vivo (re)programmed or (re)calibrated by the doctor's (external) programmer device to cope with interpatient variation, device malfunction, sensor drift, and so on.

All its benefits notwithstanding, wireless communication also opens the door to unwanted device access, which can lead to personal data theft or, worse, implant incapacitation [Halperin et al. 2008]. Since modern IMDs are embedded computing systems in actuality, specially targeted computer viruses or malware could infect implantable devices and use them to spread, potentially damaging a large patient base [Gasson 2010]. Public awareness on the matter is gradually catching up [Leavitt 2010; Strydis et al. 2008] and successful, in vitro attacks on implantable devices have already been demonstrated [Halperin et al. 2008; Li et al. 2011] to verify that such fears are not unwarranted. The attacks on the Epilepsy Foundation website—where seizures were induced in some epileptic patients by replacing website content with rapidly flashing images [Poulson 2008]—have led to the certainty that malicious attacks on IMDs are simply a matter of time and, therefore, that security is required in order to protect the steadily increasing number of patients carrying wireless IMDs.

Due to the safety-critical nature of IMDs, an implant requires a **robust security protocol** that does not allow access to individuals without knowledge of a secret password. At the same time, the protocol has to be **accessible** enough to allow unrestricted access to any first-aid responder with medical expertise in case of patient (e.g., heart attack, seizure) or device (e.g., malfunction, low battery) emergency.

Trading high-security for high-accessibility levels is just one of many conflicting requirements a designer has to take into account. Another one is **device autonomy**: Most IMDs use a battery as their energy source. In order to prolong battery life and, in so doing, avoid life-threatening and costly battery replacement (through surgery), IMDs primarily need to be ultra-low-power (ULP) devices. This poses serious limitations on the number of security protocols applicable to implants. Robust security protocols typically are prohibitively taxing on battery power.

For coping with the unique set of implant requirements in ultra-low power and ultra-high safety, smart security schemes such as "zero-power defense" [Halperin et al.

2008] (to be discussed later on) and profiling studies on implant-friendly, symmetric-encryption algorithms [Strydis et al. 2008] have already appeared in the literature. Although more studies on the various security aspects of implants are needed, we believe that the design and implementation of a complete system for IMD security is much needed. Such an IMD system architecture is expected to provide adequate security (i) without compromising the functionality and performance of the IMD, (ii) without exceeding the maximum peak power, and (iii) while maintaining a sufficiently low energy overhead.

In line with the aforementioned challenges, in this work we present (i) a new **system architecture** and (ii) a matching **secure communication protocol** for shielding IMDs against various pertinent attacks. We also design, implement, and optimize (iii) a low-power **security processor** for IMDs that executes the communication protocol, and we provide detailed performance, power, energy, and area results of the complete system. The security processor is called **Smart-Implant Security Core (SISC)** and is designed to work independently from the primary implant module; in this case, a previously published artificial-pancreas module has been selected. The SISC does so by scavenging RF power from any active wireless communication link for initiating secure data exchange, rather than wasting the implant battery.

Concisely, with this work we make the following novel contributions:

—We identify multiple potential IMD *attack scenarios* and propose a *secure protocol* for communication between the IMD and an external reader;
—We propose a new implant *system architecture* that guarantees security and safety through our security protocol and by completely decoupling the security and implant application tasks in terms of processing and powering needs;
—We provide a new, low-power *security ASIP and ANSI C-compiler* for facilitating the security protocol. Use of an ASIP executing instructions has the advantage of allowing complete updating of the protocol even after device manufacturing and implantation; and
—We demonstrate the feasibility of our solution by presenting synthesis-based results of a new proof-of-concept system implementing the secure IMD architecture and identify the trivial overheads it introduces in terms of energy and device resources.

The rest of the article is organized as follows: Section 2 presents background information and an overview of related works in the field. In Section 3, we introduce the new system architecture, detailing the proposed security protocol and the SISC architecture. Since the IMD security topic is far from being established, in Section 4 we discuss a few educated assumptions made to complete this work, we present the experimental setup used, and, finally, we report on the overheads introduced in terms of performance, area cost, and power and energy consumption of the new, secure IMD system. Overall conclusions are drawn in Section 5.

2. BACKGROUND AND RELATED WORK

Compromising implant functionality and privacy can have serious repercussions for the device host. These can range from *blackmail* and *social segregation* (for instance, if an employer can identify that a potential employee carries an IMD, he might feel inclined to not hire that person) to *treatment prevention* and even *death*. Securing an IMD against such attacks is, thus, a major challenge. There are, however, more challenges that need to be addressed in the process: Halperin et al. [2008] draw attention to the tradeoff between security, on one hand, and device resources, accessibility, and usability, on the other. In this section, we will provide background information and present related works dealing with these challenges in IMD security.

Table I. Typical Security System Components and Various Implementations Thereof
Entity authentication and message integrity can be replaced by a message authentication protocol. Components in bold are adopted in this work.

Key management	– **Initial key distribution (always needed).** – **Offline distribution system** / online distribution system (distribution of session key to authorized users). – Replacement / **no replacement**.

Entity authentication	– One-side authentication (2-way challenge-response protocol). – Mutual auth. (3-way challenge-response protocol). – Trusted party.	**Message authentication**	– **MAC (if the two sides trust each other)**. – Digital signature (if the two sides don't trust each other).
Message integrity	– Message echoing (as verification). – Hash function.	**Freshness**	– Counter. – **Random Number (RN) generation**.

Confidentiality	– **Symmetric (or private) cipher encryption** / asymmetric (or public) cipher encryption.

2.1. Security Components

Every security system can be divided into the following components [van der Lubbe 1998]:

—**Key Management** involves the generation, distribution, and (periodic) replacement of keys used for encrypting the message;
—**Entity Authentication** identifies if a message originates from a trusted entity;
—**Message Integrity** confirms a message has been received correctly; and
—**Confidentiality** prevents from disclosing a message to unauthorized entities.

These components can be implemented in a number of ways, as shown in Table I. Message authentication (MAC) protocols provide both entity authentication and message integrity. Obviously, there does not exist a single secure communication protocol that can be efficient for every system. For each system, the most suitable protocol should be selected with respect to the available resources and constraints as well as to the attack types to which the system is susceptible. An extensive study of the various alternatives in Table I has been presented in Siskos [2011] and the interested reader is encouraged to refer to his work. In this article, we will focus on selecting the components best suited for building the envisioned secure IMDs, to be presented in Section 3.

2.2. IMD Switching Modes and Access Control

While the components in Table I guarantee device *security*, provisions must also be in place to guarantee implant *accessibility* and *safety* [Halperin et al. 2008]. Accessibility plays a key role in emergency situations such as a heart attack or falling unconscious. In such situations, it is crucial that IMD security mechanisms do not prevent treatment. It is clear that *secure access* is required during normal mode of operation and *fail-open access* during emergencies. In this section, we present and discuss recently proposed methods for allowing unsecured access during emergency situations.

It has been suggested by Schechter [2010] that the encryption key is tattooed onto patients using ultraviolet micropigmentation near the scar where the IMD is implanted, making it available in case of emergency. The main drawbacks of such a tattoo are (i) the need for a blacklight to be always present during an emergency so as to read the key, (ii) a marginally increased risk of key theft, and (iii) cumbersome key replacement. Furthermore, a patient survey has shown that tattooing the key, using regular or ultraviolet ink, is generally disliked and may lead to stigmatization [Denning et al. 2010].

Switching between secure and emergency modes can also be achieved by using a magnetic switch. In this technique proposed by Halperin et al. [2008], the security protocol is active as long as the magnetic switch is on. When it is turned off (by physically swiping a magnet extradermally over the implant), unsecured access is enabled. However, an adversary may have significantly powerful equipment at their disposal capable of toggling the security mode, thus bypassing the security of this approach [Halperin et al. 2008] as well. Furthermore, systems using this technique may be prone to accidental mode switching when in the presence of relatively strong magnetic fields.

Distance-bounding-based access control has been proposed by Rasmussen et al. [2009]. Security is achieved by using (i) a credential held by the reader, which shares a secret key with the IMD, and (ii) the proximity-based protocol. During normal mode of operation, the private key is used for allowing secure access to the IMD. In emergency mode, the proximity-based protocol is used: Anyone in close proximity to the implant ($< 10cm$) can have access to it. By having the valid range in emergency mode be much smaller than in normal mode, the risk of the implant being attacked is significantly reduced. However, recent work in the RFID field has shown that current distance-based protocols are vulnerable to distance-hijacking attacks [Cremers et al. 2012] and, therefore, we question the security of distance-bounding protocols for IMDs.

A number of access schemes rely on an external device to provide accessible security to the IMD. Such a device could, for example, be hidden in the form of a necklace or watch. In the IMD-Cloaker [Denning et al. 2008], IMDGuard [Xu et al. 2011], Amulet [Sorber et al. 2012], and Personal-Security-Device [Pournaghshband et al. 2012] approaches, the external device acts as a security middleman, preventing secured access to the IMD when the two are in close proximity. If the external device is physically distanced from the IMD, fail-open access ensues. In the IMD-Shield approach [Gollakota et al. 2011], the incoming and outgoing messages to the IMD are jammed, preventing all access to the IMD while the jammer is present (i.e., only emergency mode is facilitated). This approach requires no modifications to the IMD, making it ideal to enhance already implanted IMDs with security. The main drawback of the aforementioned approaches is the vulnerability of the IMD when the cloaker or jammer is not present. For example, the device may be lost, stolen, or simply forgotten.

Criticality-aware systems change their mode of operation based on monitoring a set of critical variables [Gupta et al. 2006]. Halperin et al. [2008] have proposed to use criticality awareness for IMDs, where the IMD itself determines if the patient is in an emergency situation based on physiological parameters and provides open access in case of emergency. Hei et al. [2010] have proposed to use the patient's heartbeat as a biometric trigger; for example, the IMD may switch to emergency mode when the heart rate is above (or below, in case of cardiac arrest) a certain threshold. While criticality awareness may seem promising, there is yet to be found a (set of) parameter(s) that unambiguously capture an emergency-only situation.

2.3. Low-Power Security Techniques

Having in mind that the aforementioned works address security, accessibility, and safety mechanisms for IMDs, we should also pay attention to the overheads that they may introduce, be they in the form of performance, chip area, or—most importantly—energy consumption.

Lean security protocols tailored to IMDs have already been presented by Beck et al. [2011] and Hosseini-Khayat [2011]. Both protocols propose symmetric encryption using ciphers of low computational complexity. Besides, symmetric (block) ciphers are considered to have lower computational complexity (and, thus, power and energy requirements) compared to asymmetric ones (which feature simpler key management)

[Martin Feldhofer and Wolkerstorfer 2004]. While both proposed protocols promise low-power security, actual power results and cryptanalysis, which are essential when proposing a new security protocol, are not provided. Furthermore, we consider both protocols prone to battery Denial-of-Service (DoS) attacks, as no extra precautions are taken to prevent invalid requests from draining the implant battery.

In an attempt to combine low energy costs with device usability, Halperin et al. [2008] have proposed three zero-power techniques against radio-based IMD attacks: (i) zero-power notification for patients, (ii) zero-power authentication, and (iii) zero-power symmetric-key exchange, utilizing the first two techniques. Even though key management is missing from the schemes, these three defenses are based on RF energy harvesting on the side of the IMD. This is achieved by adding an RFID-like module to the IMD and making sure the techniques maintain scant power budgets. This last work introduces the interesting idea of RFID-style energy harvesting for performing security-related computations in implants.

The work presented in this article differs from Halperin et al. [2008] as follows:

(1) The protocol described in Halperin et al. [2008] provides one-way authentication and, after the reader has been authenticated, allows (unsecured) communication between the reader and IMD. In contrast, we provide mutual authentication in order to prevent spoofing attacks on the reader and facilitate encryption throughout the session, preventing message eavesdropping or alternation.

(2) In Halperin et al. [2008], RC5 [Rivest 1995] is suggested as the encryption algorithm. A recent survey on ciphers for implantable devices [Strydis et al. 2008] has shown that the MISTY1 encryption algorithm [Ohta and Matsui 2000] is superior to RC5 in terms of power consumption and is thus our choice for encryption.

(3) Halperin et al. [2008] present a prototype using an RFID-like module (WISP) for executing the security protocol using a standard 16-bit RISC microprocessor [Smith et al. 2006]. In contrast, we have designed a 32-bit ASIP processor that has been optimized for executing the MISTY1 cipher in order to minimize the performance and energy consumption overheads. Moreover, using an ASIP processor allows us to change the used security protocol and cipher in case their security is compromised.

(4) In Halperin et al. [2008], a zero-power defense against DoS attacks is proposed by having the IMD operate on RF-harvested energy prior to authentication. However, no actual implementation is presented. In this article, we propose, implement, and evaluate a system architecture capable of providing such zero-power defenses. Specifically, we demonstrate zero-power security is achievable by powering it through an RF link without affecting the main implant functionality.

Secure, wireless communication at minimal energy cost is a well-studied problem in RFID tags and Wireless Sensor Networks (WSNs) [Ko et al. 2010; Lee et al. 2010]. Although IMDs share similar security requirements, there are various key differences setting them apart. In this discussion, relevant ones are (i) **emergency mode**: IMDs require an emergency mode, which bypasses regular security (seen in Section 2.2), and (ii) **power harvesting**: IMDs may harvest energy from the human body through, among others, piezoelectric and thermodynamic effects, reducing or removing the need for battery-powered operations [Olivo et al. 2011]. An example of an energy-harvesting implantable device is presented in Nazhandali et al. [2005], where an ultra-low-power intraocular pressure sensor, consisting of a pressure sensor, processor, and memories, is powered by converting temperature gradients within the eyeball to electricity. Wireless IMDs may also harvest power over a generic RF link or through magnetic coupling (as many commercial devices do at the moment) of more than $\sim 25\mu W$, which is the maximum attainable with most commercial RFID antennas [FCC 2003]. This results in a typically higher power budget for the *mission-critical* IMDs compared to either the *disposable, cheap* RFID tags or the *physically remote or inaccessible* sensor motes, where security strength is often exchanged for power reduction [Cam et al. 2003; Rieback et al. 2008].

Due to these significant differences, we do not consider security protocols and mechanisms typically used in the RFID and WSN fields to be directly applicable to IMDs. On the other hand, designing a security protocol from scratch is generally not common practice as it requires a long time to fully test, to perform cryptanalysis on it, and so on. In Section 3.3, we will revisit this dilemma and propose a suitable solution.

3. SYSTEM ARCHITECTURE FOR SECURE IMPLANTS

Achieving security for an implant can be divided into two parts. The first part, present in every secure system, is to make the system resilient to attacks. The second part, not so common in other secure systems, is to make the implant very power and energy efficient. Excess power or energy consumption as well as unauthorized access could lead to compromising the implant host's health and even to death. In this section, we will go through the process of identifying potential IMD attacks, selecting the system architecture, the secure communication protocol, and the ISA of the new security core, in order to deliver a secure yet energy-efficient and versatile system.

3.1. IMD-Related Attacks

To build a secure IMD, we should start by listing the possible attacks that can be mounted. IMDs appear to be susceptible to the following array of attacks:

(1) **Entity impersonation**: Impersonation of the implant or of a reader through, for example, a replay attack. A replay attack is a network attack whereby a valid data transmission is maliciously or fraudulently repeated or delayed. Such an attack may be used for preventing treatment by, for example, scrambling the order of packets arriving at the IMD or, worse, allowing full access to the IMD by repeating previously intercepted access credentials.
(2) **Message alteration**: Altering (malicious or accidental) of the message contents during communication. Altering may be used to inject malicious and potentially dangerous commands to the IMD.
(3) **Message eavesdropping**: Eavesdropping on the messages, thus compromising sensitive patient data. Note that the privacy of this data has to be guaranteed by any device handling medical data under the Health Insurance Portability and Accountability Act[1] (US) and European Convention on Human Rights Article 8 (EU).
(4) **Battery DoS**: Battery DoS occurs when the adversary indirectly causes the battery to discharge. For instance, if one repeatedly requests a specific operation from the implant, it will repeatedly run the same authentication protocol for analyzing the request and, eventually, deplete its power source, even if the request does not result in authentication.
(5) **Jam DoS**: This type of DoS occurs when the adversary blocks the communication channel of an implant by repeatedly sending it valid or invalid messages.
(6) **Insider attack**: When the attacker is a trusted party with proper access credentials to the device. For instance, the patient's doctor can alter information in the IMD memory log to hide evidence of malpractice.

Moreover, any DoS attack may interfere with an implant's main functionality by having it spend time on handling the incoming requests. All of these attacks may, to a certain extent, pose a threat to an IMD, yet in this work we will address attacks (1) through (4). In an IMD, the communication channel will be used infrequently and mostly to communicate non-mission-critical information to a base station; even

[1] US Pub. Law 104-191, est. 1996.

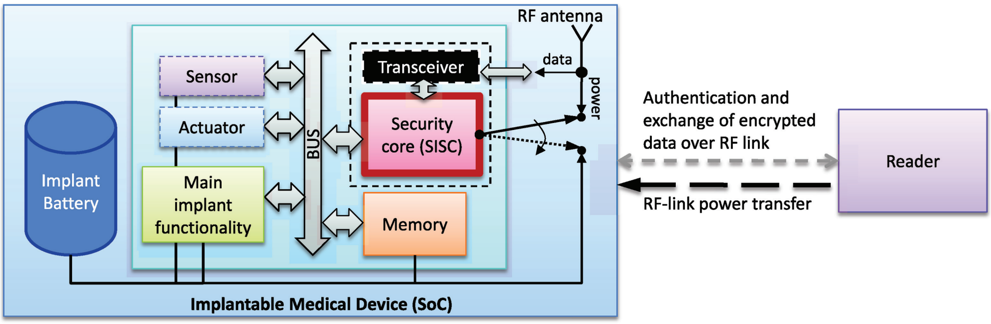
RF antenna
data
power
Transceiver
Security core (SISC)
Sensor
Actuator
Main implant functionality
BUS
Memory
Implant Battery
Implantable Medical Device (SoC)
Authentication and exchange of encrypted data over RF link
RF-link power transfer
Reader

Fig. 1. Proposed IMD system architecture, facilitating decoupled computations and powering. Secure communication is handled exclusively by the SISC core, preventing data transfers from stalling the life-critical main functionality of the implant. The SISC is powered by harvesting energy over the RF link, preventing unauthorized wireless communication requests from draining the implant battery.

treatment changes in most cases do not have a tight window within which to be performed. Therefore, the IMD will find minimal, if any, harm from a Jam DoS attack (5), which relies on blocking the channel. Furthermore, we consider the prevention of insider attacks (6) to be outside the technical scope of this article as prevention is typically achieved by security management (e.g., restricting key access to trusted parties) rather than a security protocol. For now, we assume that the doctor is always a *trusted* person and a Jam DoS attack shall *never* occur.

We plan to address attacks (1) to (3) through use of a suitable security protocol and attack (4) (battery DoS) through use of a suitable IMD system architecture. The system architecture and the security protocol are strongly linked. A decision taken for the former can complicate or simplify the implementation of the latter, and vice versa. This means that both components have to be developed concurrently. However, for clarity purposes, we describe our proposed system architecture and, thus, tackle the battery DoS problem first.

3.2. IMD System Architecture

We design a novel IMD system, the first one capable of preventing battery DoS attacks using an RF link to harvest energy from and communicate to the reader. Thereby, incoming requests do not drain the implant battery. Furthermore, we propose that the implantable system shall be partitioned into two modules—one supporting the primary IMD functionality and the other exclusively handling security; this enables us to prevent regular attacks while dealing with (DoS) attacks without interrupting the main IMD functionality.

A top-level view of the proposed System-on-Chip (SoC) architecture is presented in Figure 1. The IMD SoC consists of a number of *memory-mapped modules* connected onto a common bus. The main module is responsible for performing the primary implant functionality, such as the signal-processing task of a pacemaker. Smart biosensors (e.g., QRS-complex sensing) and bioactuators (e.g., heart pacing) are needed to interface to the living tissue. A shared-memory block is used for the rest of the modules to log or exchange implant data and wireless communication events. Last, a custom security ASIP called SISC (Smart-Implant Security Core), which is tightly coupled to a transceiver module, is responsible for running the IMD security protocol. The SISC core has its own (private) instruction (IMEM) and data memory (DMEM) blocks; the SISC specifics will be revisited in Section 3.4. The shared-memory block and the SISC memories are turned off most of the time in order to save energy. As such, it is necessary that the shared memory and the SISC IMEM are nonvolatile and have low power consumption. Therefore, Flash memories have been chosen, which are becoming increasingly compatible with CMOS logic over recent years [Shukuri et al. 2001]. In this way, the security protocol (i.e., software program running on the SISC) as well as the (prestored) encryption keys will not be lost every time RF-induced power is interrupted.

As Figure 1 illustrates, all modules are exclusively powered by a battery pack with the exception of the SISC, which can be also powered by energy harvested through the RF antenna, in a fashion similar to RFID tags. The idea behind the dual powering system of the SISC is as follows: When a new communication request is made by an external reader—authorized or unauthorized—the implant antenna harvests RF energy and powers up the previously sleeping SISC (through the transceiver module). Over the same RF link, the SISC receives an encrypted request packet and attempts to authenticate the other party. If it succeeds (i.e., this communication request has not been an attack of any sort), then the SISC knows for certain that the reader is authentic, at which point it switches the power source from the harvested RF energy to the implant battery. It is allowed to do so because it does not run the risk of a battery DoS attack but, rather, services a legitimate task: It is expected to perform some "high" energy-consuming operations, such as multiple memory accesses, data encryption and broadcasting, change of implant operation mode (e.g., firmware update), and so on. Once the power source has toggled, the SISC transmits back a similarly encrypted response packet that the reader can use, in turn, to authenticate the implant and actual data exchange can take place, as will be explained next, in Section 3.3. Obviously, if the actual operations involved afterward are not too energy taxing, the SISC could maintain its RF-harvested power source (as opposed to switching to battery mode), but we are addressing here the problem in the general case. What is crucial for our analysis is that the proposed SISC core should be able to operate within the power margin the RF link provides.

In case the opening communication request is, in fact, an attack, then the SISC would power up at zero-energy cost (i.e., through the RF-harvested energy), fail to authenticate the reader, drop the reader request, and, subsequently, go back to sleep without discharging the implant battery or disrupting the primary implant operation at all. The reasons for proposing a partitioned IMD system architecture are, thus, becoming obvious: First, the security module should not use the same resources as the main functionality module of the implant. It is not acceptable for the main functionality program (it is, often, life critical) to stall because of running the security program (*computation decoupling*). Second, we do not want the security part to drain the main implant battery (*power decoupling*) on any unauthorized communication request.

3.3. Secure Communication Protocol

Having dealt with the energy issue, we now move to the next important issue, which is bypassing security in case of emergency. In literature, we found a number of approaches to providing emergency access, as presented in Section 2.2. Out of these current alternatives, we believe that use of a magnetic switch that allows fail-open access while in close proximity to the implant is sufficient for next-generation IMDs.

In the previous section, we addressed attack (4) through the proposed IMD system architecture. In this section, we will present in detail a security protocol that complements the system design choices and addresses the remaining attacks (1), (2), and (3). There are two main approaches to designing a security protocol: (i) design of a protocol from scratch or (ii) design of a protocol based on an international standard. If a new protocol is proposed (option i), it might suffer from security vulnerabilities even when a rigorous security analysis is conducted together with the proposal. For instance, over the last years many RFID authentication protocols followed this approach and a significant number of them have been partially or completely broken [van Deursen and Radomirovic 2008; Zhuang et al. 2013; Bogari et al. 2012]. Such a trend cannot be tolerated in the implant domain due to the high criticality of such devices. On the other hand, the design of protocols conforming to a standard (option ii) facilitates the usage of these protocols in commercial devices. Standards have been deeply scrutinized by the community before being part of a regulation, which guarantees their security and low chance of latent errors [Boyd and Mathuria 2010; Bauer and Juerjens 2008]. Therefore, in this work we have based our security protocol and used primitives on standards and well-established international regulations. Under this approach, the construction of messages is set by the standard(s), but the message content and various extra fields are dependent on the application context. Last but not least, the usage of a standard security scheme for implants has the added benefit of narrowing the gap between research and actual, commercial IMDs.

Let us now revisit Table I, which presents possible choices for every layer of a security system. In the context of medical implants and with the previous system choices in mind, we have drawn final decisions for each layer of our system, appearing in the table in boldface text. First, we have decided to use a *symmetric cipher* for message encryption and for Message-Authentication-Code (MAC) calculation. The reason is that, generally speaking, a proven symmetric algorithm is sufficiently secure, yet significantly *less computationally intensive*, thus less energy consuming compared to an asymmetric one [Martin Feldhofer and Wolkerstorfer 2004; Lee et al. 2010]. Furthermore, asymmetric approaches typically require large memory footprints or expensive coprocessors like the cloaker [Denning et al. 2008]. On the other hand, symmetric systems require that each entity pair has a unique key for maintaining communication confidentiality. Consequently, as the number of communicating entities increases, storage requirements also increase. To address the issue, the assumption is made here that only a small number of valid readers (a typical example would include *three*: doctor, patient, close relative of the patient) exist. This assumption is realistic since in health care applications the number of authorized readers is small in comparison to other applications such as RFID-based access systems.

In a symmetrical system, the **key distribution** can be done by a distribution center. For our system, it is obligatory to use an *offline* distribution system, because an online one would require communication with the distribution center, which is quite impractical for IMDs. Moreover, in our case, there are only about three readers, all with their own security privileges and their own key. Therefore, there is no need for a more sophisticated key distribution scheme. Finally, replacement of the key is not needed unless a reader is compromised.

The next step is the selection of the entity authentication and message integrity methods. We have decided to cover both by using **MAC** for the data blocks. Freshness and protection against *replay attacks* are guaranteed by the use of *random numbers*. A replay attack could lead an adversary to reply information and supplant a legitimate entity. In this protocol, both entities (reader and implant) contribute with a fresh nonce, which is randomly generated in every new session, ensuring uselessness of old messages (replay attack protection). Random numbers are generated on the fly during protocol execution and only the values linked with the current session are kept in memory, thus requiring minimal memory overhead.

As previously mentioned, instead of designing a security protocol from scratch, our proposed solution is based on standards and international security recommendations. More precisely, ISO/IEC 9798 Part 2 specifies six schemes based on symmetric encryption algorithms [ISO 1999], providing various degrees of authentication: unilateral authentication, mutual authentication, and authentication with key establishment using a third entity (server). Our proposed scheme is based on the fourth protocol of this standard, as we require mutual authentication between the reader (R) and the implant (IMD). Both entities share a key K_{RI}, and its identifiers are ID_R and ID_I, respectively. Nx symbolizes a random number generated by entity X. $[[M]]_K$

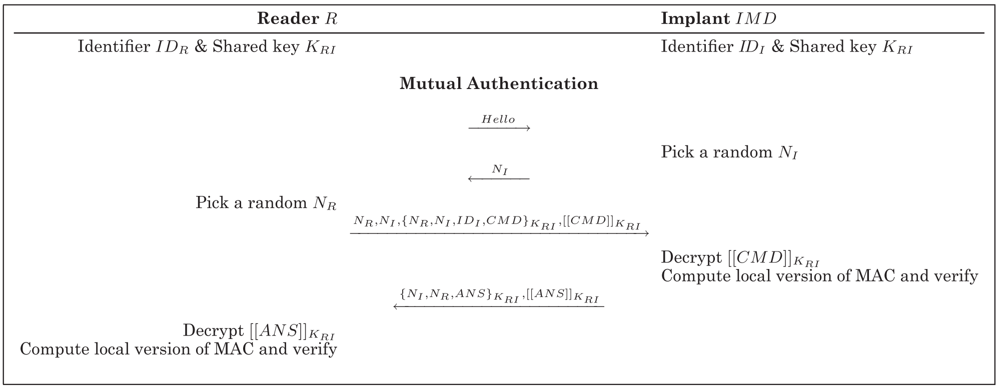
Reader R
Implant IMD
Identifier ID_R & Shared key K_{RI}
Identifier ID_I & Shared key K_{RI}
Mutual Authentication
$\xrightarrow{Hello}$
Pick a random N_I
$\xleftarrow{N_I}$
Pick a random N_R
$\xrightarrow{N_R, N_I, \{N_R, N_I, ID_I, CMD\}_{K_{RI}}, [[CMD]]_{K_{RI}}}$
Decrypt $[[CMD]]_{K_{RI}}$
Compute local version of MAC and verify
$\xleftarrow{\{N_I, N_R, ANS\}_{K_{RI}}, [[ANS]]_{K_{RI}}}$
Decrypt $[[ANS]]_{K_{RI}}$
Compute local version of MAC and verify

Fig. 2. The proposed IMD security protocol conforming to ISO/IEC 9798, which guarantees mutual authentication between the reader and IMD. Minimal effort is required from the IMD prior to reader authentication, allowing the IMD to operate on RF-harvested energy.

represents the encryption of message M with key K to provide confidentiality and $\{M\}_K$ symbolizes the encryption of message M with key K to provide confidentiality and integrity. The messages may include a command field (CMD) or answer field (ANS), which are dependent on the application. The exchanged messages, shown in Figure 2, in our four-pass, mutual authentication protocol are described in the following steps:

(1) $R \rightarrow IMD$: *hello*. The reader wakes up the implant.
(2) $IMD \rightarrow R$: N_I. The implant generates a random number and sends this nonce to the reader.
(3) $R \rightarrow IMD$: $N_R, N_I, \{N_R, N_I, ID_I, CMD\}_{K_{RI}}, [[CMD]]_{K_{RI}}$. The reader generates a random number and computes a MAC. This MAC includes the random number received and the one generated on board and the identifier of the target IMD (ID_I). Additionally, a command field (CMD) is included as a part of this message. Finally, these two random numbers together with the MAC and an encrypted version of the command ($[[CMD]]_{K_{RI}}$) are sent to the implant.
(4) $IMD \rightarrow R$: $\{N_I, N_R, ANS\}_{K_{RI}}, [[ANS]]_{K_{RI}}$. The implant decrypts the CMD received, computes a local version of the MAC ($\{N_R, N_I, ID_I, CMD\}_{K_{RI}}$), and checks its equality with the received value. Note that only the target implant knows the identifier (ID_I) used and the two nonces associated with the session. If this validation fails, the implant aborts the protocol. If not, the IMD allows the SISC to switch to the battery supply and sends a MAC, which includes the two aforementioned nonces and the response command (ANS). Additionally, an encrypted version of the answer ($[[ANS]]_{K_{RI}}$) is attached to this message.
(5) R: First, the reader decrypts the answer. Then, knowing the ANS field and the two nonces linked to the current session, the reader calculates a local version of the MAC ($\{N_I, N_R, ANS\}_{K_{RI}}$). If the received values and the computed values are equal, the reader and the implant are mutually authenticated. If not, the protocol returns an answer.

Since the proposed scheme is based on a well-known International Standard, its security has already been analyzed in depth. Moreover, recent studies like Boyd and Mathuria [2010] do an extensive analysis of security schemes and recommend 9798-2 as a secure entity authentication scheme.

3.3.1. Implementation Aspects. We start by fixing the bit length of the values used in the protocol by taking as reference common values used in low-cost RFID tags due to their rough similarities with implants (computational, memory, power consumption restrictions) [Halperin et al. 2008; Juels 2006]. The identifiers are 96 bits long ($|ID_R| = |ID_I| = 96$) and the bit length of random numbers is set to 32 ($|N_R| = |N_I| = 32$) [EPCglobal Inc]. Commands are 32 bits long ($|CMD| = 32$), as we expect that this will be large enough to cover all command options for the IMD, and answer codes are a multiple of 64 bits ($|ANS| = n \times 64$). These lengths have been chosen to align the request and answer sizes to the 64-bit block size of our chosen cipher, discussed next.

Regarding *primitives*, we discard public cryptography due to the scarce resources of IMDs. Our symmetric system is mainly based on the use of a lightweight and secure block cipher, abbreviated CIPH, using a shared key (K). From all possible candidates—and without loss of generality—we choose *MISTY1,* which is a 64-bit lightweight block cipher designed by Ohta and Matsui [2000]. Our choice is based on a suitability analysis on symmetric ciphers for medical implants by Strydis et al. [2008]. Out of 13 profiled ciphers, MISTY1 has been found best suited for implants since it ranks high across most metrics such as power consumption, energy cost, and encryption speed.

Aiming to reuse the selected primitive, we also use a MAC algorithm that is based on a symmetric key block cipher. This cipher-based MAC is abbreviated **CMAC**. Our algorithm follows the NIST 800-38B Recommendation [NIST 2005], providing guidelines for using block ciphers for our purposes. Moreover, for subkey generation we follow what is dictated in the specification (i.e., NIST 800-38B; pages 7–8) [NIST 2005]. The subkeys (K_1 and K_2) are generated and stored in the memory of the entities involved (e.g., reader and implant) at the key distribution phase. To compute the MAC of message M (i.e., $\{M\}_K$), M is divided into blocks of 64 bits: $M_1||M_2||\dots M_m$, where $m = |M|/64$. The CMAC algorithm is described here:

CMAC algorithm compliant with NIST 800-38B
1. $C_0 = 0^b$
2. For $i = 1$ to m, let $C_i = CIPH_K(C_{i-1} \oplus M_i)$
3. $T = C_m$
4. Return T

Therefore, the MAC of message M is T (i.e., $T = \{M\}_K$). Apart from the MAC, random numbers are used in the protocol. We opt for a standard approach again: As specified in the NIST 800-38A specification [NIST 2001] ("Recommendation for Block-Cipher Modes of Operation"), we use our block cipher in a counter mode, called CTR. The current value of the counter is denoted by T_j and RN represents the outputted 32-bit random number. After each nonce generation, the counter value (T_{j+1}) is updated. The initial value of the counter is set at the key distribution phase (i.e., $T_0 = random\_seed$). The algorithm proposed is described here:

Block cipher-CTR mode (compliant with NIST 800-38A)
1. $O_j = CIPH_k(T_j)$
2. $R_N = |O_j|_{0\cdots31}$
3. $T_{j+1} = |O_j|_{32\cdots64}$

After determining the security protocol, we proceeded with its implementation in ANSI C and compilation by the SISC C-compiler. The program starts and ends with the creation and destruction of the keys. Note that, in realistic systems, the key creation and destruction functions should run only once because the software of the protocol should not end after accomplishing a single communication session. This program binary is set to run on every new power-up of the SISC triggered by an external communication request.

3.4. SISC Architecture

In this section, we concisely present the SISC architecture, a custom-designed security ASIP. By definition, an ASIP is more power consuming but also more *flexible* compared to an ASIC through changing the binary code it executes. Thus, an ASIP strikes a better tradeoff between optimized performance/power and design flexibility.

The use of processor-based implant controllers has been demonstrated in various designs in the past [Stotts et al. 1989; Harrigal and Walters 1990; Fernald et al. 1991; Wouters et al. 1994; Valdastri et al. 2004; Wang et al. 2004; Jalilian et al. 2004] whereby in vivo reprogramming of the implant has been successfully performed. Although these cases focused on reprogramming the main implant processor, in this work we extend the reprogramming capabilities of the implant security core. We believe this feature to be pivotal to the design of realistically safe and secure future implants for the following reason: IMDs make up a narrow market niche and, as such, have to this point attracted only limited (malevolent) attention. As IMD use is becoming more widespread over time, more systematic and versatile security attacks are bound to be mounted. Unfortunately, under such conditions, the incorporated security of many future commercial IMDs will be compromised. The only eventual safeguard against this will be the in vivo reprogramming of such devices for adding extra phases in the security scheme, changing the protocol altogether, and so on. Such major updates are impossible under an ASIC approach and device replacement through surgery will be needed, which is an action to be taken only as a last resort.

Since our SISC design needs to be low energy but also perform nontrivial computations, we have chosen to implement an integer, in-order RISC-style architecture. We have selected to design a typical five-pipeline-stage processor, which consists of Instruction-Fetch, Decode, Execution, Memory, and Write-Back stages. It consists of a 16-bit instruction set along with 16 registers of 32 bits each. In our case, the use of 32-bit registers is important because the encryption and decryption algorithms of our benchmark are operating on 32-bit quantities. When run on a standard, in-order RISC core, the encryption algorithm generates a significant amount of read-after-write hazards. In order to reduce the instruction binary size—or, equivalently, the implant memory needs—in our SISC design, NOPs are removed from the binary at compile time and are dynamically inserted at runtime whenever a hazard is detected. With this optimization, the IMEM size is $24KB$ and the DMEM size is $16KB$.

The SISC ISA consists of 25 instructions listed in Table II. Profiling of the SISC security code traces has yielded various potential optimizations. One such optimization is the replacement of the very frequent *mov-and-beqz* group by an instruction extension (*mandb*), highlighted in gray in the table. With this optimization introduced in the baseline core, the execution time has been reduced by 34.31% and the energy cost by 34.26% at a scant power increase of 0.73%. The SISC processor and C compiler have been designed by using the Synopsys Processor-Designer and Compiler-Designer tools.

4. EXPERIMENTAL RESULTS

Having described all components of the secure IMD system, we can now move to the evaluation of our design. Compared to the current state of the art in IMDs, our SoC employs *additional* resources—namely, the SISC running security tasks and a number of required modifications in the system interconnect, the memory sizes, the power lines, and so on—to provide security. A direct comparison between an unsecured, *reference IMD SoC* and our security-enabled and, thus, more resource-costly SoC is needed for identifying the overheads our design incurs in terms of performance, area, power, and

Table II. SISC Instruction-Set Architecture
Supported types and syntax are listed. Instruction extension is highlighted in gray.

type	**instruction format**
rr	op [4] reg [4] reg [4] funct [4]
rrr	op [4] reg [4] reg [4] reg [4]
ri	op [4] reg [4] imm [8]
jump	op [4] imm [12]

name	**type**	**assembly**	**action**
jr	rr	jr rd	branch to addr[rd]
and	rr	and rd,rs	rd ← rd and rs
lw	rr	lw rd,rs	rd ← mem[rs]
sw	rr	sw rd,rs	mem[rs] ← rd
mov	rr	mov rd,rs	rd ← rs
not	rr	not rd,rs	rd ← not rs
or	rr	or rd,rs	rd ← rd or rs
xor	rr	xor rd,rs	rd ← rd xor rs
sub	rr	sub rd,rs	rd ← rd-rs
add	rr	add rd,rs	rd ← rd+rs
lb	rr	lb rd,rs	rd ← mem[rs] (byte)
se	rr	se rd,rs	if(rd==rs) rd←1 else rd←0
sgt	rr	sgt rd,rs	if(rd>rs) rd←1 else rd←0(signed)
sgtu	rr	sgtu rd,rs	if(rd>rs) rd←1 else rd←0(unsigned)
beqz	rr	beqz rd,rs	if(rd==0) branch to addr[rs]
subi	ri	subi rd,imm	rd←rd-imm
addi	ri	addi rd,imm	rd←rd+imm
li	ri	li rd,imm	rd←imm
sftl	ri	sftl rd,imm	rd←rd<<imm
sftr	ri	sftr rd,imm	rd←rd>>imm(sign extension)
sftru	ri	sftru rd,imm	rd←rd>>imm(zero extension)
cb	rrr	cb rd,rs1,rs2	exchanges the $rs2^{th}$-byte of rd by the LSB of rs1
mandb	rrr	mandb rd,rs1,rs2	if (rd and rs1==0) branch to addr[rs2]
j	jump	j imm	branch to addr[imm]
jal	jump	jal imm	branch to addr[imm] and r15←PC+1

energy costs. Moreover, to evaluate our solution on even terms, a secure reference SoC has to also be established for making comparisons.

Existing commercial IMDs typically implement their functionality around a single core, which often is an embedded, off-the-shelf DSP running the entire implant application software. If security would be added, this main module would also have to be burdened with running security-related tasks such as data encryption, MAC calculation, and so on. Therefore, we define as our reference design an SoC where sensor, actuator, and memory modules exist but *no special security module* is included. In its unsecured version, the main module of the SoC runs only the main functionality of the IMD, while in its secure version, it additionally runs the secure communication task.

Our experimental setup consists, then, of two *proof-of-concept* SoC designs: (i) a single-core, reference SoC with and without execution of security tasks, and (ii) our dual-core, secure SoC including the SISC module. Without loss of generality, for the main functionality module—common across all three SoC instances—a *realistic* implant case study has been used: the so-called ‘artificial pancreas (abbrev. AP) as discussed in the next section.

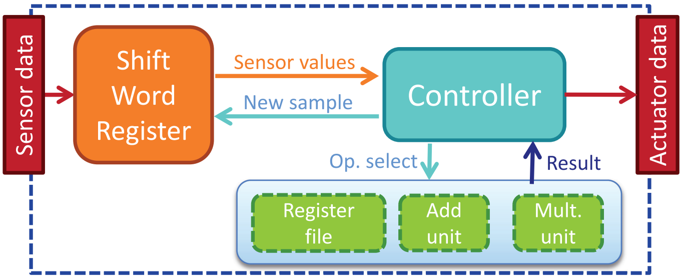

Fig. 3. Simplified block diagram of the Biostator-II artificial-pancreas hardware controller. On every new actuation event, a shift word register storing the five most recent glucose measurements (sensor values) is consulted to compute the amount of insulin to be released (actuator values).

4.1. Case Study: Biostator II

An AP implant is a next-generation, yet simple in conception, device that allows for *closed-loop control* of the blood glucose levels of diabetic patients. It essentially consists of a glucose sensor (e.g., H_2O_2 Platinum-needle detector), a control unit, and an insulin actuator (e.g., a micropump insulin injector) to close the loop. For our working case, we have adopted the Biostator II implementation by Pagkalos et al. [2011].

Biostator II is an ultra-low-power ASIC implementation of an AP controller. The main advantage of an implanted controller, compared to manual insulin injections or insulin pumps, is a more reactive control of the insulin injection rate, resulting in improved treatment and, of course, a more discreet and convenient alternative.

The Biostator II (ASIC) module is depicted in Figure 3. At every minute, the module powers up and receives a new blood glucose value, storing it in a local memory block (Shift-Word Register, SWR) containing the five most recent values. After storing this value, a small state machine is executed, which takes 42 clock cycles to calculate the insulin release rate based on the five stored blood-glucose values using a number of additions and multiplications. Biostator II represents the main implant functionality in Figure 1 and communicates to its glucose sensor and insulin actuator through two bus interfaces. While these interfaces have been implemented, the actual sensor and actuator modules, as well as the IMD transceiver, are analog components outside the scope of this article. Instead, three simple buffers for producing and consuming physiological data, respectively, have been used as scaffolds in the SoC.

For our experiments, the proposed IMD SoC can integrate the ASIC block of the Biostator II as the main functionality module. However, the reference design requires a software version of the application to be running in the main module. Therefore, except for an optimized version of the original ASIC design, a *C version* of the Biostator II engine has also been implemented and validated against its hardware counterpart.

For both the software and the hardware version of the Biostator II, the shared memory of the IMD is used for logging sensor data, actuator data, wireless communication events (e.g., what configuration changes have been applied), and intercore communication (e.g., parameter changes in the Biostator II). Based on typical glucose and insulin trends [Daly et al. 1998; Pagkalos et al. 2011], storing the sensor and actuation data every 5 minutes for a maximum of 72 hours (3 days) is sufficient for reconstructing the patient's recent blood insulin and blood glucose trends. Based on these numbers we have estimated that an 8kB shared-memory block will be large enough to store all recent sample values (in single-floating-point precision) and device access history (i.e., timestamps).

Each log entry is 96 bits long (16 bits for the sensor and actuator values, 32 bits for a timestamp), resulting in 96 bytes of data stored per hour, or 6.75kB for a

3-day timespan. In order to maintain device access history, we log the 25 most recent successful reads and 25 successful implant configuration changes. Each memory entry will contain the reader ID (96 bit), command (32 bit), and timestep (32 bit), resulting in a total of around 1kB of data. The resulting memory size of the shared memory is, thus, 8kB. Due to the limited SISC memories and performance (see Section 4.2), we impose a maximum of 2kB to be read out by a reader per request.

The IMD can be accessed wirelessly by both the patient (e.g., for monitoring the blood glucose levels) and his or her treating physician (e.g., for configuring parameters of the Biostator II module, performing sensor recalibration, or collecting logged physiological data). In our case study, we assume that a patient is frequently monitoring his or her glucose levels (every 8 hours is quite typical with diabetics) and will ask for a full glucose/insulin history since the last read (i.e., 8 hours of data). It should be stressed that the AP case is used here without loss of generality to better illustrate the working of a secure IMD.

4.2. Synthesis Results

Having outlined our experimental base, we can now present detailed synthesis results of the various SoC instances and the new SISC module in particular. First, we will describe the findings for the SISC core and protocol to demonstrate the feasibility of the energy-harvesting security strategy, after which we will extend our discussion to the complete SoC. Since the Synopsys Processor Designer and Compiler Designer have been used for designing the SISC, the tools have been also employed for generating a simple 32-bit, five-pipeline-stage, in-order RISC core and C compiler for use as the main functionality module for the reference SoC designs. The core supports no advanced (micro)architectural features such as bypassing and branch prediction, thus exhibiting a low area cost (to be further discussed in Section 4.2.2).

Besides, we have employed the Synopsys Design Compiler to synthesize all IMD modules and Synopsys PrimeTime in order to get accurate power measurements. Due to availability constraints, the modules have been synthesized in *UMC* $90nm$ *CMOS–Standard Performance (SP)* technology. More suitable libraries, for example, UMC Low-Leakage (LL) or UMC Low-Power (LP), were not available to us. Power measurements have been performed by feeding the IMD modules with data typical to the case study and extracting a signal switching-activity file (SAIF) in ModelSim. This activity file is—along with the synthesized netlist—fed to PrimeTime in order to determine the actual system power consumption.

As the employed UMC library does not support Flash memories, we have estimated the overheads of the shared memory and SISC-IMEM based on synthesis results of the SISC-DMEM (SRAM) and known SRAM-to-Flash ratios [Park et al. 2003]. Furthermore, the sensor, actuator, and wireless transceiver modules are not part of this evaluation, although they are part of the actual IMD; they are all mixed-signal components that cannot be synthesized with the UMC CMOS library and, besides, are outside the scope of the current article.

It is further worth noting that the Biostator II ASIC and software task have been supplied with typical glucose values by a simulated glucose sensor, while the SISC core runs the IMD side of the security protocol considering successful and unsuccessful external reader accesses. Last, the system runs at $20MHz$, which is a typical operating frequency for modern implants [Strydis 2011].

4.2.1. SISC. In Table III, for each IMD-side protocol stage, a detailed synthesis report of each SISC subsystem is given; per subsystem the *active per-transaction time*T_{trans}^{ON} (i.e., the per-transaction duty cycle of the subsystem) and the *per-transaction energy*

Table III. Breakdown of Security Protocol Execution Times and Energy Costs (per Transaction) When Run on the SISC
Two distinct execution phases are shown, one based on scavenged RF power and one based on IMD battery power.

SISC-protocol steps	**ANS size: 2***					**ANS size: 288****				
	T^{ON}_{trans}		E^{ON}_{trans} (uJ)			T^{ON}_{trans}		E^{ON}_{trans} (uJ)		
	$(msec)$	**core**	**IMEM**	**DMEM**	**total**	$(msec)$	**core**	**IMEM**	**DMEM**	**total**
RF link power										
Step 2 (per trans.)	2.7	0.55	0.03	0.91	1.50	2.6	0.55	0.03	0.91	1.50
Step 4 - Preauth. (per trans.)	10.9	2.11	0.12	3.72	5.95	10.9	2.11	0.12	3.72	5.95
Battery power										
Step 4 - Postauth. (per trans.)	8.2	1.59	0.09	2.81	4.49	789.3	153.20	8.88	270.56	432.63
Total RF harvested (per trans.)	13.6	2.66	0.15	4.63	7.45	13.5	2.66	0.15	4.63	7.45
Total battery powered (per trans.)	8.20	1.59	0.09	2.81	4.49	789.30	153.20	8.88	270.56	432.63
Total battery powered (per day)	**24.6**	**4.77**	**0.28**	**8.43**	**13.47**	**2367.9**	**459.59**	**26.64**	**811.67**	**1297.90**

Note: One day contains three legitimate transactions.
*ANS size 2: e.g., reply to command “calibrate sensor”.
**ANS size 288: e.g., reply to command “fetch data of last 8 hours.”

consumption E^{ON}_{trans} are reported. Per-transaction energy figures of the complete SISC are also reported.

Note that, in the table, we have chosen to illustrate the costs for executing a *single wireless transaction,* which is the minimal, atomic operation that the SISC can perform. We report on two transaction instances of the system: one transaction that entails the MAC and a short ANS package of two 32-bit words ($m = 2$), for example, a simple acknowledgment of receiving the command—and another that entails the MAC and a long ANS package of two hundred and eighty-eight 32-bit words ($m = 145$), which is the typical amount of data requested periodically by the reader in the presented AP case (see Section 4.1).

As explained for the security protocol in Section 3.3, the IMD is responsible for executing steps 2 and 4 of the protocol (the rest being executed by the reader). On the basis of the power source used, we split the IMD-side protocol execution into two main phases: reader *preauthentication* and *postauthentication*. During the preauthentication phase, the SISC wakes up on RF power and executes step 2 and part of step 4 of the algorithm (verification of the reader). For these tasks, the whole SISC tile consumes a mere E^{ON}_{trans} *of* $7.45\mu J$ regardless of the ANS size and the authenticity of the request, as shown in Table III (“Total RF harvested (per trans.)”). Based on this result and on the FCC guidelines [FCC 2003], we can conclude that the power and energy budgets demanded in the preauthentication phase of our secure IMD are *within* the capabilities of current general RF-harvesting techniques (transceiver energy costs included). Consequently, the proposed SISC architecture offers *zero-battery-power defense* to unauthorized readers.

In the absence of the SISC and security protocol—as is the case with currently available commercial implants—the $7.45\mu J$ reported in Table III would be the **actual energy penalty** an *unsecured* IMD would have to pay every time it suffered a battery DoS attack. In such a case, the unsecured IMD would wake up on battery power, process the illegal request, drop it, and power down, the whole process taxing its battery a nontrivial $7.45\mu J$. Since thousands of such attacks can be mounted quite easily, the crippling effect to the implant lifetime is straightforward.

Contrary to the preauthentication phase, the *total* E^{ON}_{trans} for the postauthentication phase (second part of step 4 of the protocol) is command dependent. For the AP case, it can range from $4.49\mu J$ (for a small ANS) to $432.63\mu J$ (for a large ANS) per transaction. Therefore, the energy overhead of security lies in the micro-Joule range and is, thus, reasonably low for battery operation (see Section 4.2.3 for a viability discussion).

4.2.2. System architecture. In Table IV, a breakdown synthesis report of three different SoCs is presented:

(1) an *unsecured single-core* reference SoC running the Biostator II software;
(2) a *secure single-core* reference SoC running the Biostator II software; and
(3) our *proposed secure SoC,* which includes the ASIC Biostator II module and the customized SISC running the security protocol software.

For each of these cases, Table IV reports the per-module *area A* and *average active power consumption* P^{ON}_{avg} (i.e., the power consumed when the module is turned on). Note that the table also reports a *full-day's battery-energy drainage* E^{ON}_{day}, which, in the case of our proposed IMD SoC, is independent of the number of battery DoS attacks mounted against the implant due to its zero-power authentication cost. Thus, the figures reported in the table for our IMD SoC account for the energy costs when running on battery power. That is, preauthentication energy costs (up to the second half of step 4 in the security protocol) are not included in these figures (and can be seen in Table III). Note also that, per day, a nominal number of three legitimate (wireless) reads is assumed, each transferring 8 hours of implant data (*ANS*: 288 words), as discussed in Section 4.1.

Table IV. Per-Component Breakdown of SoC Area, Average Active Power, Active Time, and Active Energy Cost (per Day) When Running on *Battery* for the Reference System (with and without Security) and the Proposed Secure SoC

Reference SoC w/o security				
Component	$A\ (\#Cells)$	$P_{avg}^{ON}\ (uW)$	$T_{day}^{ON}\ (msec)$	$E_{day}^{ON}\ (uJ)$
Baseline RISC - core (Biostator II S/W)	54558	240.40	344.26	82.76
Baseline RISC - IMEM	8kB*	3.75	344.26	1.29
Baseline RISC - DMEM	1152	2.51	344.26	0.86
Interconnect	6111	87.30	0.22	0.02
Shared memory	8kB*	3.75	0.04	$\sim$0.00
Total SoC	>**61821**	–**	–**	**84.93**

Reference SoC w/ security				
Component	$A\ (\#Cells)$	$P_{avg}^{ON}\ (uW)$	$T_{day}^{ON}\ (msec)$	$E_{day}^{ON}\ (uJ)$
Baseline RISC - core (Biostator II S/W)	54558	240.40	344.26	82.76
Baseline RISC - core (Security S/W)		224.60	4711.19	1058.13
Baseline RISC - IMEM	32kB*	15.00	5055.45	75.83
Baseline RISC - DMEM	157232	342.80	5055.45	1733.01
Interconnect	6111	87.30	0.35	0.04
Shared memory	8kB*	3.75	0.09	$\sim$0.00
Total SoC	>**217901**	–**	–**	**2949.77**

Secure SoC (including SISC)				
Component	$A\ (\#Cells)$	$P_{avg}^{ON}\ (uW)$	$T_{day}^{ON}\ (msec)$	$E_{day}^{ON}\ (uJ)$
SISC - core	56038	194.15	2367.90	459.59
SISC - IMEM	24kB*	11.25	2367.90	26.64
SISC - DMEM	157232	342.80	2367.90	811.67
Biostator II (ASIC)	28650	190.30	0.09	0.02
Interconnect	8843	94.40	0.35	0.04
Shared memory	8kB*	3.75	0.09	$\sim$0.00
Total SoC	>**250763**	–**	–**	**1297.96**

Secure vs Reference SoC w/o security	>**405%**	–**	–**	**1528%**
Secure vs Reference SoC w/ security	<15%	–**	–**	–56%

*Actual cell numbers for estimated Flash memories not available, thus, excluded from the total area.
**Not meaningful for the complete SoC (in particular for power, a total value cannot be calculated as it depends on the tasks running at any point in time).

The difference between the first and second reference design is that the latter, on top of the Biostator II task, also runs the security task (based on our selected security scheme as shown in Section 3.3). For doing so, the baseline-RISC IMEM and DMEM subsystems need to be increased in size to accommodate for the increase in binary size, thus resulting in a more than 3.5× increase in area. The interconnect and shared-memory area use remains unaffected since, in both cases, the amount of on-chip traffic and sensor data handling, respectively, stays the same.

In terms of daily activity (T_{day}^{ON}), running the security scheme adds approximately 4.7 seconds of execution time to the baseline RISC. This, combined with the increased RISC memory sizes, results in 35× higher costs in total daily energy consumption (E_{day}^{ON}). It is interesting, though, that the RISC-core P_{avg}^{ON}, when running the security task, is slightly lower than that of the Biostator II task, meaning that running the security task in the reference SoC will not lead to an increase in average system power consumption.

These overheads in area, execution time, and energy allow us to appreciate the costs introduced to an IMD when secure functionality is desired. The costs remain nontrivial even through use of a very lean security scheme such as the one selected in Section 3.3.

This observation supports our original claim that borrowing security schemes from other application fields that match IMD requirements poorly can potentially lead to largely inefficient secure IMDs.

Having discussed naive implementation of security in IMDs (i.e., the secure, single-core, reference SoC), we can now compare that design with our proposed, dual-core, secure SoC. As Table IV reveals, our new secure SoC design comes at a less than 15% increase in area and at a 56% decrease in daily energy compared to the secure reference SoC. Therefore, our secure SoC offers high security levels at additional chip-area costs that are negligible with respect to the total implant package and at an energy budget of less than $\sim 1.3mJ$ for one full day of nominal operation. It is noteworthy that the energy-hungrier, secure reference SoC is still susceptible to battery DoS attacks.

Besides, in our SoC instance, the Biostator II software task has been replaced by its more efficient hardware (ASIC) counterpart as an extra optimization. While this optimization is beneficial for the IMD, it is not a prerequisite for implementing our security design. As can be seen from the table, this conversion saves about $85\mu J$ of daily energy in total, which would lead to a dual-core, secure SoC with still significant energy reductions (47%) compared to the secure reference SoC. Besides, with respect to the unsecured, reference SoC, our dual-core, secure SoC requires approximately 4× more area and 6.3× more energy to provide security.

With regard to P_{avg}^{ON}, the SISC core exhibits an average power profile ($194.15\mu W$) similar to the Biostator II ASIC, and lower than its respective RISC-core profile ($224.60\mu W$), revealing the optimized functionality of the SISC. This can help reduce the chance of hotspots in the implant chip. The SISC IMEM (Flash) is low power and the SISC DMEM (SRAM) is the main power culprit (~41%) in the SISC tile and the SoC, for that matter. This means that employing a lower-power SRAM block is bound to improve power figures considerably.

4.2.3. Discussion. The previous analysis of our system has spawned two important results: (i) Although it is a proof-of-concept system, **our proposed SoC achieves zero-energy defense**. That is, the SoC requires only $7.45\mu J$ for powering up the SISC and accepting or rejecting a reader request. This budget is very feasible to harvest through existing RF link technology. (ii) **Our proposed SoC achieves low-overhead security communication**: there is no *performance overhead* in the primary implant functionality due to the delegation of security tasks to the SISC. What is more, even under very high (for a ULP device) authenticated data transmissions (two hundred and ninety 32-bit words), the *energy overhead* that the security subsystem imposes to the SoC is less than $1.3mJ$.[2] In their extensive survey on IMDs, Daniluk and Niewiadomska-Szynkiewicz [2012] specify that a typical ICD consumes $\sim 25\mu J$ per pacing pulse, thus requiring an energy budget of $\sim 2.16J$ per day under normal circumstances (i.e., without the occasional delivered ICD shock) and is known to last for 5 to 7 years before the battery needs replacement. This practically means that, for a commercial-grade IMD that adopts our security enhancements, there will not be *any noticeable drop* in the lifetime of the implant battery since the security-imposed energy overhead ($< 1.3mJ$) is three orders of magnitude lower than the baseline implant energy budget of $2.16J$.

5. CONCLUSIONS

With this work we have addressed several challenging issues regarding the design and implementation of a secure IMD. Our aim has been to provide high security levels at low

[2] The transceiver energy overhead caused by the extra transmission of the MAC block has been estimated and is in the range of a few micro-Joules, thus negligible.

energy consumption while not jeopardizing the primary implant functionality. To this end, we have, first, proposed a novel system architecture, where a security ASIP (SISC) decouples the security tasks from the implant functionality. We have combined this system with a carefully selected new security scheme, based on known power-sensitive security standards. This security scheme offers defense against common attacks such as entity impersonation, message alteration, and message eavesdropping. The system architecture complements security by allowing the SISC to consume RF-induced power while dealing with unauthenticated readers, thus providing additional defense against the more sophisticated battery DoS attacks. In order to estimate the feasibility of our system architecture, SISC core, and security protocol, we have implemented and analyzed a complete proof-of-concept system architecture featuring an artificial-pancreas controller (Biostator II) as the main implant module. The analysis has verified that our prototype requires very low energy to perform reader authentication ($7.45\mu J$), effectively implementing the previously suggested zero-battery-energy defense mechanism through pure RF link energy harvesting. What is more, our prototype guarantees zero performance overheads in running both the main implant tasks and the security tasks. Our analysis has revealed that the daily energy cost of our proposed SoC architecture ($1.3mJ$), which supports the main IMD functionality and useful security tasks under high-volume data transmission, is negligible for the energy budget of commercially available chronic implants. Compared to a reference single-core IMD, which runs the same security protocol but is still not able to deal with battery DoS attacks, our secure SISC-enabled SoC requires 15% more area and consumes 56% less energy on a daily basis.

ACKNOWLEDGMENTS

This work has been supported by the EU-funded project DeSyRe (grant agreement no: 287611). It would not have been complete without the invaluable feedback of Dr. J. v/d Lubbe and Dr. I. Broustis (security), Ir. D. Riemens (synthesis), and A. L. Mansano (RF energy harvesting). We would also like to acknowledge the Brain Innovations Consortium (www.braininnovations.nl) and the constructive criticism of Dr. S. Isaza.

REFERENCES

Bauer, A. and Juerjens, J. 2008. Security protocols, properties, and their monitoring. In *Proceedings of the 4th International Workshop on Software Engineering for Secure Systems (SESS'08)*. ACM, New York, NY, 33–40.

Beck, C., Masny, D., Geiselmann, W., and Bretthauer, G. 2011. Block cipher based security for severely resource-constrained implantable medical devices. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*. ACM, 62.

Bogari, E. A., Zavarsky, P., Lindskog, D., and Ruhl, R. 2012. An analysis of security weaknesses in the evolution of RFID enabled passport. In *Proceedings of the 2012 World Congress on Internet Security (WorldCIS'12)*. 158–166.

Boyd, C. and Mathuria, A. 2010. Protocols for authentication and key establishment. In *Information Security and Cryptography* 3rd Ed. Springer Publishing Company, Incorporated.

Cam, H., Ozdemir, S., Muthuavinashiappan, D., and Nair, P. 2003. Energy efficient security protocol for wireless sensor networks. In *Proceedings of the Vehicular Technology Conference*. Vol. 5. IEEE, 2981–2984.

Cremers, C., Rasmussen, K. B., Schmidt, B., and Capkun, S. 2012. Distance hijacking attacks on distance bounding protocols. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'12)*. 113–127.

Daly, M. E., Vale, C., Walker, M., Littlefield, A., Alberti, K., and Mathers, J. C. 1998. Acute effects on insulin sensitivity and diurnal metabolic profiles of a high-sucrose compared with a high-starch diet. *American Journal of Clinical Nutrition 67,* 6, 1186–1196.

Daniluk, K. and Niewiadomska-Szynkiewicz, E. 2012. Energy-efficient security in implantable medical devices. In *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS'12)*. IEEE, 773–778.

Denning, T., Borning, A., Friedman, B., Gill, B. T., Kohno, T., and Maisel, W. H. 2010. Patients, pacemakers, and implantable defibrillators: Human values and security for wireless implantable medical devices. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*. 917–926.

Denning, T., Fu, K., and Kohno, T. 2008. Absence makes the heart grow fonder: New directions for implantable medical device security. In *Proceedings of the 3rd Conference on Hot Topics in Security*. USENIX Association, 5:1–5:7.

EPCglobal Inc. 2008. Class-1 Generation-2 UHF RFID protocol for communications at 860 MHz-960 MHz (version 1.2.0).

FCC. 2003. *MICS Medical Implant Communication Services*. FCC 47CFR95.601-95.673 Subpart E/I. Rules for MedRadio Services.

Fernald, K., Cook, T., III, T. M., and Paulos, J. 1991. A microprocessor-based implantable telemetry system. *IEEE Computer 24*, 23–30.

Gasson, M. N. 2010. Human enhancement: Could you become infected with a computer virus? In *Proceedings of the IEEE International Symposium on Technology and Society (ISTAS'10).* 61–68.

Gollakota, S., Hassanieh, H., Ransford, B., Katabi, D., and Fu, K. 2011. They Can Hear Your Heartbeats: Non-invasive Security for Implantable Medical Devices. Retrieved December 2, 2013 from http://groups.csail.mit.edu/netmit/IMDShield/paper.pdf.

Gupta, S. K., Mukherjee, T., and Venkatasubramanian, K. 2006. Criticality aware access control model for pervasive applications. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications*. 251–257.

Halperin, D., Heydt-Benjamin, T. S., Fu, K., Kohno, T., and Maisel, W. H. 2008. Security and privacy for implantable medical devices. *IEEE Pervasive Computing 7*, 30–39.

Halperin, D., Heydt-Benjamin, T. S., Ransford, B., Clark, S. S., Defend, B., Morgan, W., Fu, K., Kohno, T., and Maisel, W. H. 2008. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy.* IEEE Computer Society, 129–142.

Harrigal, C. and Walters, R. 1990. The development of a microprocessor controlled implantable device. In *Proceedings of the 1990 16th Annual Northeast Bioengineering Conference*. 137–138.

Hei, X., Du, X., Wu, J., and Hu, F. 2010. Defending resource depletion attacks on implantable medical devices. In *GLOBECOM*. IEEE, 1–5.

Hosseini-Khayat, S. 2011. A lightweight security protocol for ultra-low power ASIC implementation for wireless implantable medical devices. In *Proceedings of the 5th International Symposium on Medical Information & Communication Technology (ISMICT'11)*. IEEE, 6–9.

ISO. 1999. Information technology–Security techniques–Entity authentication–Part 2: Mechanisms using symmetric encipherment algorithms, ISO/IEC 9798-2:2008. International Standard.

Jalilian, E., Turner, L., Jullien, G., and Mitchev, M. 2004. Design of an implantable multichannel neurostimulator for restoring impaired gastrointestinal motility. In *Proceedings of the 9th Annual Conference of the International FES Society*.

Juels, A. 2006. RFID Security and privacy: A research survey. *IEEE Journal on Selected Areas in Communications 24,* 2, 381–394.

Ko, J., Lu, C., Srivastava, M. B., Stankovic, J. A., Terzis, A., and Welsh, M. 2010. Wireless sensor networks for healthcare. *Proceedings of the IEEE 98,* 11, 1947–1960.

Leavitt, N. 2010. Researchers Fight to Keep Implanted Medical Devices Safe from Hackers. *Computer 43*, 11–14.

Lee, J., Kapitanova, K., and Son, S. H. 2010. The price of security in wireless sensor networks. *Computer Networks 54,* 17, 2967–2978.

Li, C., Raghunathan, A., and Jha, N. K. 2011. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In *Proceedins of the 13th IEEE International Conferenc on e-Health Networking Applications and Services (Healthcom'11)*. 150–156.

Martin Feldhofer, S. D. and Wolkerstorfer, J. 2004. Strong authentication for RFID systems using the AES algorithm. In *Cryptographic Hardware and Embedded-Systems*. Springer, 85–99.

Nazhandali, L., Minuth, M., Zhai, B., Olson, J., Austin, T., and Blaauw, D. 2005. A second-generation sensor network processor with application-driven memory optimizations and out-of-order execution. In *Proceedings of the 2005 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*. ACM, 249–256.

NIST. 2001. Recommendation for Block Cipher Modes of Operation—Methods and Techniques. NIST Special Publication 800-38a. National Institute of Standards and Technology.

NIST. 2005. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38b. National Institute of Standards and Technology.

Ohta, H. and Matsui, M. 2000. A description of the misty1 encryption algorithm. *RFC2994, November*.

OLIVO, J., CARRARA, S., AND DE MICHELI, G. 2011. Energy harvesting and remote powering for implantable biosensors. *IEEE Sensors Journal 11,* 7, 1573–1586.

PAGKALOS, I., HERRERO, P., EL-SHARKAWY, M., PESL, P., OLIVER, N., AND GEORGIOU, P. 2011. Vhdl implementation of the biostator ii glucose control algorithm for critical care. In *Proceedings of the Biomedical Circuits and Systems Conference (BioCAS'11)*. IEEE, 94–97.

PARK, C., SEO, J., BAE, S., KIM, H., KIM, S., AND KIM, B. 2003. A low-cost memory architecture with nand xip for mobile embedded systems. In *Proceedings of the 1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*. ACM, 138–143.

POULSON, K. 2008. Hackers Assault Epilepsy Patients via Computer. Retrieved from www.wired.com/politics/security/news/2008/03/epilepsy.

POURNAGHSHBAND, V., SARRAFZADEH, M., AND REIHER, P. 2012. Securing legacy mobile medical devices. In *MobiHealth*.

RASMUSSEN, K. B., CASTELLUCCIA, C., HEYDT-BENJAMIN, T. S., AND CAPKUN, S. 2009. Proximity-based access control for implantable medical devices. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*. ACM, 410–419.

RIEBACK, M., CRISPO, B., AND TANENBAUM, A. 2008. RFID Guardian: A battery-powered mobile device for RFID privacy management. In *Information Security and Privacy*. Springer, Berlin, 259–273.

RIVEST, R. L. 1995. The rc5 encryption algorithm. In *Fast Software Encryption*. Springer, 86–96.

ROGER, V. L., TURNER, M. B., ET AL. 2011. *Heart Disease and Stroke Statistics – 2011 Update: A Report from the American Heart Association*. American Heart Association.

SCHECHTER, S. 2010. Security that is Meant to be Skin Deep Using Ultraviolet Micropigmentation to Store Emergency-Access Keys for Implantable Medical Devices. In *HealthSec*. USENIX.

SHUKURI, S., YANAGISAWA, K., AND ISHIBASHI, K. 2001. Cmos process compatible ie-flash (inverse gate electrode flash) technology for system-on-a-chip. In *Proceedings of the IEEE Conference on Custom Integrated Circuits, 2001*. 179–182.

SISKOS, D. 2011. *A Co-processor for a Secure Implantable Medical Device*. M.S. thesis, Delft University of Technology.

SMITH, J. R., SAMPLE, A. P., POWLEDGE, P. S., ROY, S., AND MAMISHEV, A. 2006. A wirelessly-powered platform for sensing and computation. In *UbiComp 2006: Ubiquitous Computing*. Springer, 495–506.

SORBER, J., SHIN, M., PETERSON, R., CORNELIUS, C., MARE, S., PRASAD, A., MAROIS, Z., SMITHAYER, E., AND KOTZ, D. 2012. An amulet for trustworthy wearable mhealth. In *Proceedings of the 12th Workshop on Mobile Computing Systems & Applications*. ACM, 7.

STOTTS, L., INFINGER, K., BABKA, J., AND GENZER, D. 1989. An 8 bit microcomputer with analog subsystems for implantable biomedical application. *IEEE Journal of Solid-State Circuits*. 292–300.

STRYDIS, C. 2011. *Universal Processor Architecture for Biomedical Implants: The SiMS Project*. Ph.D. thesis, Delft University of Technology, Delft, Netherlands.

STRYDIS, C., ZHU, D., AND GAYDADJIEV, G. 2008. Profiling of symmetric encryption algorithms for a novel biomedical-implant architecture. In *Proceedings of the ACM International Conference on Computing Frontiers (CF'08)*. 231–240.

VALDASTRI, P., MENCIASSI, A., ARENA, A., CACCAMO, C., AND DARIO, P. 2004. An implantable telemetry platform system for in vivo monitoring of physiological parameters. *IEEE Transactions on Information Technology in Biomedicine*. Vol. 8, 271–278.

VAN DER LUBBE, J. C. A. 1998. *Basic Methods of Cryptography*. VSSD.

VAN DEURSEN, T. AND RADOMIROVIC, S. 2008. Attacks on rfid protocols. *IACR Cryptology ePrint Archive 2008*, 310.

VARSHNEY, U. 2003. Pervasive healthcare. *Computer 36,* 12, 138–140.

WANG, L., HAMMOND, P., JOHANNESSEN, E., TANG, T., ASTARAS, A., BEAUMONT, S., MURRAY, A., COOPER, J., AND CUMMING, D. 2004. An on-chip programmable instrumentation microsystem for gastrointestinal telemetry applications. In *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'04)*. 2109–2112.

WOUTERS, P., COOMAN, M. D., LAPADATU, D., AND PUERS, R. 1994. A low power multi-sensor interface for injectable microprocessor-based animal monitoring system. *Sensors and Actuators A: Physical*. 41–42, 198–206.

XU, F., QIN, Z., TAN, C. C., WANG, B., AND LI, Q. 2011. IMDGuard: Securing implantable medical devices with the external wearable guardian. *IEEE INFOCOM* 1862–1870.

ZHUANG, X., WANG, Z.-H., CHANG, C.-C., AND ZHU, Y. 2013. Security analysis of a new ultra-lightweight RFID protocol and its improvement. *Journal of Information Hiding and Multimedia Signal Processing 4,* 3.

Received June 2013; revised October 2013; accepted November 2013