Matched Digital PUFs for Low Power Security in Implantable Medical Devices

Teng Xu, James B. Wendt, and Miodrag Potkonjak
Computer Science Department
University of California, Los Angeles
{xuteng, jwendt, miodrag}@cs.ucla.edu

***Abstract*—Wireless communication is widely used in Implantable Medical Devices (IMDs) to facilitate data transmission, device programming, and real-time monitoring. However, wireless systems are easy targets for attackers to inspect and potentially breach. Thus, security and privacy have become principal design requirements for IMDs. The challenge in secure IMD design stems from the conflicting constraints of the IMD (e.g. utility, safety, privacy, security, size, and low power). In this paper, we propose a new ultra low power approach for non-invasive and secure communication and operation of IMDs. We exploit the inherent process variations in an FPGA's SRAM cells to create a unique and random input-output mapping, which we then match in an ASIC design, thus, creating two matched digital physical unclonable functions (PUFs). The smaller design is embedded with the IMD while the FPGA remains with the programmer (e.g. doctor). We present our architecture, introduce accompanying protocols for secure cryptographic communication and trusted remote computation, and provide an analysis of the system's resilience to various security attacks. Our system is the first hardware-based digital security system proposed for IMDs. It is orders of magnitude lower in delay and energy consumption that traditional cryptographic techniques.**

I. Introduction

The development of small form sensors along with the reducing form factor of wireless embedded sensing systems has enabled the practical application of remote monitoring the human body. These developments, coupled with medical advances in implantation techniques, have enabled the inception of the Implantable Medical Device (IMD), an intra-body embedded sensing system. Examples of IMDs include ECGs, pacemakers, and insulin pumps.

Because of the extremely remote physical environment that the IMD inhabits—the device is *remote* in the sense that it is only physically accessible via, arguably, one of the most personally pervasive procedures, surgery—it is absolutely imperative that they are designed to be highly reliable and low power consuming [1]. These design constraints ensure that the device has low maintenance requirements, thus reducing the number of invasive procedures required to replace, re-power (if battery operated), and update the system.

Due to the intra-body nature of the IMD and its physical inaccessibility to the external world, new IMD designs have incorporated wireless radios and communication protocols for the purposes of data collection and programming. Implementation of these wireless protocols has significantly contributed to the overall utility and operation of IMDs by removing the need for invasive control, monitoring, and maintenance techniques and introducing non-invasive wireless access.

However, the introduction of wireless radios in IMDs has also introduced new security design challenges [2] [3]. Prior to wireless enabling in IMDs, the devices were safe from attackers since they were, for the most part, only accessible through very pervasive means. Wireless communication, however is available to anyone with a radio within transmission range of the device. Since attacking an IMD has high potential for endangering the very life of a patient, security—particularly, securing wireless channels—has become a premier design constraint for these embedded devices [4].

Current security techniques developed for IMDs include physiological key generation [5] and wireless channel shielding or "cloaking" [6]. These techniques either rely on existing cryptographic software techniques, which are very power intensive, or require an external third device to shield the IMD. Security flaws have been found in both of these proposals by Rostami et al. [3].

Thus, we propose to use digital security primitives, in the form of physical unclonable functions (PUFs), to enable low power and robust IMD security. PUFs are low power physical digital hardware systems that have very complex but stable input to output mappings. In general, a PUF is effectively a very complex mathematical function that is easy to evaluate but impossible to predict [7]. As the name suggests, the device is also impossible to physically replicate.

Our system consists of two PUFs: (i) the intra-body integrated security circuit (small form and ultra low power), and (ii) an external device that only the programmer (e.g. doctor) has access to. The two PUFs are matched at implantation-time such that they implement the same functionality (i.e. their input-output mappings are identical). Their function remains complex and unpredictable, so as to maintain security, and we take the necessary precautions to ensure that the system is unclonable after implantation. We also propose ultra low power security protocols for authentication and cryptographic communication. Ours is the first digital IMD security system that uses orders of magnitude less power consumption than traditional cryptographic techniques.

II. Related Work

The increased prevalence of pervasive embedded health care systems, such as body area networks and IMDs, has motivated the research and development of new privacy and security solutions for these systems [8] [9]. Furthermore, the integration of wireless technologies into IMDs has made security a premier design consideration for these devices, which

978-1-4799-5701-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICHI.2014.12

are arguably, the most pervasive and life-critical of embedded wireless health systems.

Successful attacks on wireless-enabled IMDs have been demonstrated and discussed [10]. Li et al. even demonstrated exposure of private patient information (e.g. glucose levels) during IMD communication [4]. Proposed solutions for IMD security include key generation techniques that take advantage of the randomness in physiological values [11]. A key agreement scheme was proposed by Hu et al. [5], however was later shown to have security flaws by Rostami et al. [3].

Key sharing through non-radio channels has also been proposed. Halperin et al. demonstrated acoustic sharing of keys from the IMD to the programmer through the human body using piezoelectrics [10]. Chang et al. proposed the use of an artificial voltage signal at pico-amp currents (utilizing the body as a low frequency electric carrier) to build communication channels [12]. These techniques rely on the fact that an attacker will not come into physical contact with the patient in order to snoop the key. The actual security of such methods to eavesdropping have not yet been thoroughly validated [3].

In general, the difficulty in secure design stems from the conflicting requirements of the IMD, which are privacy, security, safety, utility, and low power [13] [14]. Current techniques are limited by the specific security primitives they are built upon. For example, software level encryption is orders of magnitude more power hungry than hardware solutions, however even hardware-based primitives, such as lightweight security protocols, only enable secret key cryptography [15] [16]. Thus, we propose to change the underlying IMD security primitive to the first digital hardware solution that is orders of magnitude smaller in energy and size than traditional techniques and that also implements public key cryptography. Our proposal is also resilient to physical and side-channel attacks and enables new protocols for authentication and secure communication.

III. Preliminaries

A. Physical Unclonable Functions

A PUF is an unclonable hardware system that has a complex but definite mapping of inputs to outputs. In essence, a PUF is a very complex mathematical function derived from the intrinsic physical behaviour of its design. The mapping of inputs to outputs must remain easy to evaluate (for the purposes of authentication) but impossible to predict (for the purposes of security). Current proposed implementations utilize process variation, device ageing, and matching techniques to implement PUFs as low power security primitives [17] [18] [19]. A set of PUF based security protocols are proposed in [20] and [21]. Moreover, many PUF-based applications such as remote sensors [22] [23], and random number generator [24] are also proposed.

B. Process Variation in FPGAs

An FPGA is an integrated circuit designed to be configured after manufacturing. FPGA architectures consist primarily of an array of configurable logic blocks containing lookup tables (LUTs). These LUTs are then configured by the programmer to implement some binary function and are routed to other parts of the FPGA so as to implement the programmer's design. The configuration of each LUT is kept in SRAM memory cells.

SRAM cells in FPGAs experience a unique phenomenon when the device is powered up: they initialize to a seemingly random assignment of values (0s and 1s) that is unique to that FPGA and is constant throughout successive power-ups [25] [26]. These values are indirectly associated with the physical properties of the SRAM cells and their accompanying circuitry. This is a result of a phenomenon known as process variation, which is defined as the deviation of integrated circuit (IC) parameters (e.g. threshold voltage and effective length) from the nominal specifications that manifest as a result of manufacturing processes [27]. In the case of the FPGA, these variations manifest themselves as race conditions at power-up which set SRAM cells to particular default values.

Since the SRAM cells contain the configurations for the LUTs in the FPGA, the binary functions that are implemented at power-up per FPGA are inherently randomized and different between all FPGAs. It is important to note that process variation manifests differently even between two identically designed FPGAs who, despite having the same netlist, will power up with radically different SRAM cell assignments and therefore, different LUT configurations. Industrial manufacturers of FPGAs usually zero all LUTs during their power-up routines, however, we keep the randomized LUTs in tact in order to take advantage of their per-FPGA unique functionality.

IV. Design Overview

The remainder of this paper describes in detail our two device digital system, enabling encrypted communication between an IMD and external programmer (e.g. doctor) that is both low power and resilient to physical and side-channel attacks. We take advantage of the inherent process variation found in an FPGA's SRAM memory cells to create a unique complex function by connecting a random selection of randomly configured LUTs. Before implantation of the IMD, we configure the lookup tables (implemented using non-volatile memory) of the intra-body IC to match that of the selected LUTs on the FPGA which are known only by the programmer.

The intra-body IC is unclonable because after implantation it is impossible to physically access and thus characterize. The only way to access it is through wireless communication and pervasive physical means. Since the FPGA LUTs are randomly configured at power-up due to intrinsic process variation, it is only by gate level characterization that the default SRAM cell values can be measured thus potentially enabling an attacker to clone the device. We eliminate this possibility by physically removing (e.g. burning) those pins on the FPGA which enable gate level characterization. We analyze the security properties of the system in Section VIII.

V. Intra-body IC

Due to the IMD's high restrictions on power and area, we develop a compact and low-power intra-body IC for secure message transmission. We first describe the architecture of the device, then analyze its randomness and security properties, and discuss its frequency and power model.

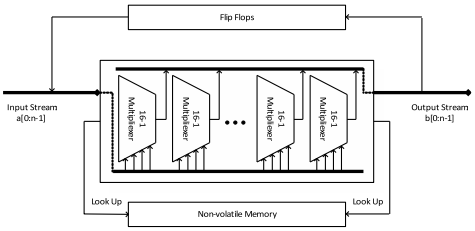


Fig. 1: Architecture of the intra-body IC.

A. Architecture

The architecture of the intra-body IC shown in Figure 1 is a sequential ASIC circuit consisting of multiplexers and non-volatile memory. The key idea is to use multiplexers to randomly choose input signals from the input vector to look up in the non-volatile memory cells in order to produce the output vector. Subsequent cycles use previously created output vectors as input. The contents of the non-volatile memory are set once, when the intra-body device is matched to the external FPGA before implantation.

Specifically, in the first cycle, a random seed (e.g. generated from physiological values) of n bits is used as the primary input. Each multiplexer randomly chooses m bits of this input vector as its input signal (Figure 1 depicts $m = 4$) to access the non-volatile memory and produce a 1 bit output. The multiplexer and the non-volatile memory together form the intra-body IC's lookup tables.

We implement the same number of multiplexers as there are bits in the input stream, n, in order to create n output bits in the current cycle. In subsequent cycles, the previously produced n-bit outputs are used as inputs. As the number of cycles increases the output stream becomes more and more random. Analysis of this randomness is discussed in Section V-B.

Note that this generic architecture allows for physical replication of the design while appointing differences in functionality to the values of the non-volatile memory cells.

B. Randomness Analysis

We construct our intra-body IC using 64 multiplexers ($n = 64$) each with 4-bit input signals. We begin the test by randomly generating a 64-bit input vector and applying it as a random seed to the device. We fetch the current output vector every 32 cycles and join it into the final output stream. The final output vector is then XORed with the current random seed to generate a new random seed for the next 32 cycles.

We implement Von Neumann correction on the final output stream to remove bias before testing the randomness of the output stream using the NIST statistical test suite [28]. The average success rates are 96% or higher for all tests and are listed in Table I.

C. Speed and Power Analysis

In order to model the power and delay characteristics of the intra-body IC, we utilize an open source 45-nm gate library to conduct table lookup-based calculations [29]. IC component switching and leakage power are based on gate sizes and load capacitances. Meanwhile, gate delay is indexed in a similar manner using the worst-case slew propagation time accounting for interconnect dependencies.

Statistical Test	Avg. Success Ratio
Frequency	100%
Block Frequency (m=128)	99.3%
Cusum-Forward	98.2%
Cusum-Reverse	98.2%
Runs	97.9%
Longest Runs of Ones	97.6%
Rank	99.9%
Spectral DFT	99.1%
Non-overlapping Templates ($m = 9$)	96.2%
Overlapping Templates ($m = 9$)	98.3%
Universal	100%
Approximate Entropy ($m = 8$)	98.5%
Rand. Excursions ($x = 1$)	99.2%
Rand. Excursions Variant ($x = -1$)	98.1%
Serial ($m = 16$)	99.0%
Linear Complexity ($M = 500$)	97.7%

TABLE I: NIST statistical test suite average success rate for outputs from the networked lookup table circuit. 1000 bitstreams of 1000 bits are passed to each test. The test passes for *P-Value*$\geq \sigma$, where σ is 0.01.

We simulate the intra-body circuit in Figure 1 using the same settings outlined in Section V-B. Table II shows the frequency, power usage, and energy consumption per bit for different supply voltages (V_{dd}). Note that energy consumption is almost negligible compared to the energy required for wireless radio data transmission, which is often orders of magnitude higher.

V_{dd} [V]	Freq. [MHz]	Power [mW]	Energy [pJ/bit]
1.5	3309	11.37	1.72
1	1611	4.74	1.47
0.8	1027	3.03	1.48
0.5	263	1.18	2.25

TABLE II: Frequency, power usage, and energy consumption of the intra-body IC under different supply voltages. The intra-body IC simulated in this experiment consists of 64 multiplexers and produces output stream every 32 cycles.

VI. Operation

In order to enable the intra-body IC to exchange messages with the programmer's external device, we match the two in such a way that both devices can produce identical output streams when given the same inputs. Therefore, we can use the shared output stream to encrypt messages in the intra-body IC and subsequently decrypt them using the programmer's device, and vice versa.

In this section we address two properties in the programmer's device related to its unclonability and its extensiveness. Unclonability ensures that even if an attacker steals the device, they can not reproduce its functionality. Extensiveness refers to the device's capability to be used as a platform for communication with a multitude of intra-body ICs using only small modifications or even no modifications at all. A general use case for this scenario might be a single doctor with one external device that can communicate securely with many

patient IMD intra-body ICs. We enable these two properties by implementing an FPGA-based SRAM PUF.

A. FPGA-based SRAM PUF

The FPGA-based SRAM PUF is based on the fact that each SRAM cell has a high probability to be initialized to either 0 or 1 at power-up due to manufacturing variability and intrinsic process variations. The power-up procedure ultimately amplifies the differences in strength of two or more transistors, setting the SRAM cell to a specific value (0 or 1). Consequently, these power-up values are random and cannot be altered by human decision.

A valid concern of this power-up procedure is that the content of the SRAM cells may not be stable for different environmental conditions. A recently proposed hot carrier injection based power-up technique has completely solved this problem and has been demonstrated in experimentation [26].

We power up the FPGA to create the SRAM PUF and use it as programmer's device platform. The configurable logic blocks which contain LUTs and flip-flops are the fundamental logic units in the FPGA. Since the contents of the LUTs are stored in SRAM cells, at power-up, the contents of the LUTs, and hence their functionality, is randomly allocated. Note that this specific FPGA can not be replicated because the process variations in each SRAM cell are unique and random. Even two FPGAs with the same netlist and structure as one another will power up with completely different configurations.

B. Device Matching

After creating the FPGA-based SRAM PUF, it is matched with an intra-body IC. Note that matching can either occur after fabrication of the IMD by the manufacturer, who we consider a trusted party, or, since the intra-body IC utilizes non-volatile memory to implement its own lookup tables, the matching task can be assigned to the doctor who would perform the matching procedure just prior to implantation.

In order to match the FPGA-based SRAM PUF with the intra-body IC, we randomly select n LUTs in the FPGA, note the contents of their SRAM cells after power-up, and allocate the same contents to the non-volatile memory in the intra-body IC. Once the functional components of each device are matched, we match the network connections between the matched LUTs in both the FPGA and the intra-body IC. Since the ASIC device cannot be rerouted, we alter the FPGA routes to match that of the intra-body IC.

Now that both devices contain the same LUT contents connected by the same network, they produce the same outputs for a given set of inputs, and according to our analysis in Section V-B, are identical random number generators.

We ensure that the intra-body device is unclonable by implanting it inside the patient. The FPGA is made unclonable by permanently and physically disabling all of the non-input and non-output pins that could potentially be used for gate level characterization. In summary, both devices are now PUFs since they are unclonable and produce stable but definite input-output mappings that are impossible to predict yet easy to authenticate (we discuss our protocols in the following section). Together, the intra-body IC and programmer's device form a matched digital security system.

Algorithm 1 Public Key Communication

1: Alice chooses a random seed s and computes $K = E_A(s)$
2: Alice computes $M = m \oplus K$
3: Alice sends M and s to Bob
4: Bob computes $K = E_B(s)$
5: Bob computes $m = M \oplus K$

Algorithm 2 Authentication

1: Alice chooses a random seed s and computes $E_A(s)$
2: Alice sends s to Bob
3: Bob computes $E_B(s)$
4: Bob sends $E_B(s)$ to Alice
5: Alice compares $E_A(s)$ and $E_B(s)$
6: **if** $E_A(s) == E_B(s)$ **then**
7: Alice authenticates Bob
8: **end if**

VII. Protocols

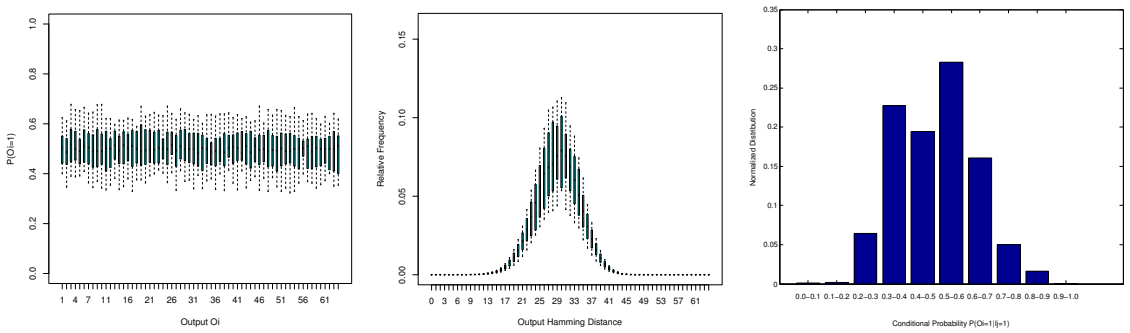
We present two protocols that utilize our matched digital PUFs: public key communication and authentication. In the following sections, we refer to Alice as the owner of the intra-body IC (e.g. a patient with the intra-body IC implanted inside of her body), and we refer to Bob as the programmer (e.g. the doctor). Furthermore, we use $E_A(x)$ and $E_B(x)$ to refer to the encryption functions of Alice's and Bob's respective PUFs as described in Section V-A; x is the input vector applied to the device. Note that since Alice and Bob match their devices before implantation, $\forall x\,(E_A(x) = E_B(x))$.

A. Public Key Communication

Public key communication is one of the most widely used and fundamental protocols for secure message exchange. The detailed steps of our IMD security system implementation are enumerated in Protocol 1. Alice sends Bob a message m such that only Bob can read it but no other party can learn any new information about m (other than its encrypted value). This protocol is important because neither the intra-body IC owner nor the external device user wants the sending message to be stolen by any third party. Note that in Protocol 1, even though the attacker could monitor and intercept the wireless channel and snoop M and s, he can not decrypt the message m since s/he does not have access to either matched device nor a characterization of either.

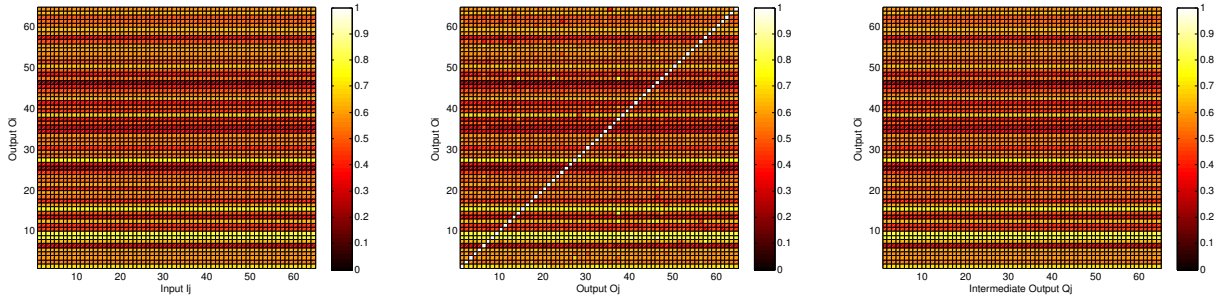
B. Authentication

Authentication is a basic cryptographic protocol that is especially important in medical systems where it is imperative that privacy and confidentiality be protected. In this protocol, Alice authenticates the party who intends to access her IMD. The authentication is based on the following two facts, (i) only a programmer's device that matches with Alice's intra-body IC can be authenticated because $\forall x\,(E_A(x) = E_B(x))$, and (ii) only Bob has the programmer's device. Bob can also confirm that any transmission sent via the IMD is really from Alice by switching their roles in Protocol 2.



(a) (b) (c)

Fig. 2: (a) Probability that an output bit is equal to 1. (b) Measuring the avalanche effect by measuring the distribution of differences between two output vectors (hamming distance) when their corresponding input vectors differ by one hamming distance. The error bars in (a) and (b) depict the maximum, 0.75 quantile, mean, 0.25 quantile, and minimum measurements. (c) Distribution of conditional probabilities $P(O_i = 1|I_j = 1)$ for all pairs of inputs, i, and outputs, j.



(a) (b) (c)

Fig. 3: Conditional probabilities between (a) output bits O_i and input bits I_j, (b) output bits O_i and output bits O_j, and (c) output bits O_i and intermediate output bits Q_j.

VIII. Security Analysis

In this section, we statistically analyze the matched system's resistance against different potential security attacks. For this analysis we use the same simulation configuration as described in Section V-B.

A. Avalanche Effect

An attacker might attempt to predict outputs using knowledge of the outputs for similar inputs. This attack is dangerous when the output vector with similar input vectors are highly correlated with one another. In cryptography, cipher diffusion is achieved if a change in the input by a small amount (e.g. one bit) results in a significant change in the output. This is called the avalanche effect.

To test if our system implements this effect, we measure the hamming distance between output vectors upon changing one bit of the input vector at each iteration. Ideally, the distribution should be in the form of a binomial distribution with the peak residing at half the total number of outputs. The result in Figure 2b matches that of a binomial distribution indicating that our system does implement the avalanche effect and is highly resilient to the related attack.

B. Frequency Prediction

An attacker can collect output data from one of the matched devices and builds a probability distribution for each output. Ultimately, the attacker attempts to predict each output O_i based on statistical distributions. The goal of the attacker is to predict $P(O_i = c)$, where $c = 0$ or 1. The ideal secure situation is that an output is 0 with a probability of 0.5. Figure 2a shows the mean value of the probability that each output bit is equal to 1. The average probability shows high tendency to be close to 0.5 which indicates resilience to this type of attack.

C. Input-based Correlation

Another type of attack attempts to build correlation mappings between an output bit, O_i, and an input bit, I_j. The goal in this attack is to predict the conditional probability, $P(O_i = c_1|I_j = c_2)$, where c_1 and c_2 are either 1 or 0. For example, if the attacker observes that output O_i is 1 a large majority of the time that input I_j is 1, and if the current input I_j is 1, then he can guess with high probability that output O_i is 1 as well. The ideal situation is that the probabilities remain 0.5. Figure 3a depicts a colormap of the conditional

probabilities, $P(O_i = 1|I_j = 1)$, in simulation for a single instance of our PUF. We also include a flat distribution of these values in Figure 2c. Both figures indicate low potential for prediction.

D. Output and Intermediate Output-based Correlation

Similar to the previously described attack, this attack attempts to predict an output bit O_i according to the value of a corresponding output bit O_j or even an intermediate output bit Q_j. In the first case, if two output bits have a strong correlation, then the attacker can deduce the output vector by knowing a subset of the output bits. In the second case, we assume the attacker has somehow obtained some intermediate results (e.g. output bits in cycle 16), possibly through a side-channel attack, then attempts to predict the final outputs (e.g. output bits in cycle 32) based on that mapping. We present a conditional probability map of $P(O_i = 1|O_j = 1)$ in Figure 3b and $P(O_i = 1|Q_j = 1)$ in Figure 3c depicting low potential for a prediction-based attack using output to output and intermediate output to output correlations.

IX. Conclusion

The introduction of wireless technology to IMDs has presented new challenges in secure IMD design. These challenges derive from the conflicting constraints of the IMD (e.g. utility, reliability, security, size, and power). Traditional heavy weight cryptographic techniques are not suited for such an environment due to their often high power demands and slow execution.

Thus, we have proposed a matched digital PUF system that enables non-invasive, secure, and ultra-low energy cryptographic communication between an IMD and the programmer. Our system is designed to be unclonable and is implemented in digital hardware so that it is fast, low power, and resilient to environmental and operational conditions. Furthermore, we analyzed the resilience of the system to various potential attacks and demonstrated new protocols for authentication and public key communication.

X. Acknowledgement

This work was supported in part by the NSF under Award CNS-0958369, Award CNS-1059435, and Award CCF-0926127, and in part by the Air Force Award FA8750-12-2-0014.

References

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

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

[3] M. Rostami, W. Burleson, A. Juels, and F. Koushanfar, "Balancing security and utility in medical devices?" *Design Automation Conference*, pp. 13, 2013.

[4] C. Li, A. Raghumathan, and N. K. Jha, "Hijacking an insulin pump: security attacks and defenses for a diabetes therapy system," in *IEEE International Conference on e-Health Networking Applicaitons and Services*, pp. 150-156, 2011.

[5] C. Hu et al., "OPFKA: secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," *INFOCOM*, pp. 2274-2282, 2013.

[6] F. Xu, Z. Qin, C. Tan, B. Wang, and Q. Li, "IMDGuard: securing implantable medical devices with the external wearable guardian," *INFOCOM*, pp. 1862-1870, 2011.

[7] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 170-178, 2002.

[8] S. Warren et al., "Interoperability and security in wireless body area network infrastructures," *IEEE Engineering in Medicine and Biology Society*, pp. 3837-3840, 2005.

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

[10] D. Halperin et al., "Pacemakers and implantable cardiac defibrillators: software radio attacks and zero-power defenses," *IEEE Symposium on Security and Privacy*, pp. 129-142, 2008.

[11] 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," *Parallel Processing Workshop*, pp. 432-439, 2003.

[12] S.-Y. Chang, Y.-C. Hu, H. Anderson, T. Fu, and E. Huang, "Body area network security: robust key establishment using human body channel," *HealthSec*, pp. 5-5, 2012.

[13] D. Halperin, T. Kohno, T. Heydt-Benjamin, K. Fu, and W. Maisel, "Security and privacy for implantable medical devices," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 30-39, 2008.

[14] W. H. Maisel and T. Kohno, "Improving the security and privacy of implantable medical devices," *New England Journal of Medicine*, vol. 362, no. 13, pp. 1164, 2010.

[15] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," *ICCAD*, pp. 670-673, 2008.

[16] P. Yalla, and J-P. Kaps, "Lightweight cryptography for FPGAs," *International Conference on ReConFigurable Computing and FPGAs*, pp. 225-230, 2009.

[17] M. Potkonjak, S. Meguerdichian, A. Nahapetian, and S. Wei, "Differential public physically unclonable functions: architecture and applications," *Design Automation Conference*, pp. 242-247, 2011.

[18] T. Xu, J. B. Wendt, M. Potkonjak, "Digital Bimodal Function: An Ultra-Low Energy Security Primitive," *ISLPED*, 2013.

[19] T. Xu, M. Potkonjak, "Robust and Flexible FPGA-based Digital PUF," to appear in *FPL*, 2014.

[20] N. Beckmann, M. Potkonjak, "Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions," Information Hiding: 11th International Workshop 2009, pp. 206-220, Darmstadt, Germany, 2009.

[21] J. B. Wendt, M. Potkonjak, "The Bidirectional Polyomino Partitioned PPUF as a Hardware Security Primitive," IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2013.

[22] M. Potkonjak, S. Meguerdichian, J.L. Wong, "Trusted Sensors and Remote Sensing," *IEEE Sensors*, pp. 1104-1107, 2010.

[23] J. B. Wendt, M. Potkonjak, "Nanotechnology-Based Trusted Remote Sensing," *IEEE SENSORS*, pp. 1213-1216, October 2011.

[24] T. Xu, M. Potkonjak, "Lightweight digital hardware random number generators," *IEEE SENSORS*, pp. 1-4, 2013.

[25] D. Holcomb, W. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1198-1210, 2009.

[26] K. Miyaji, T. Suzuki, S. Miyano, and K. Takeuchi, "A 6T SRAM with a carrier-injection scheme to pinpoint and repair fails that achieves 57% faster read and 31% lower read energy," *Solid-State Circuits Conference Digest of Technical Papers*, pp. 232-234, 2012.

[27] S. Borkar et al., "Parameter variations and impact on circuits and microarchitecture," *Design Automation Conference*, pp. 338-342, 2003.

[28] A. Rukhin et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," National Institute of Standards and Technology, Special Publication 800-22, rev. 1a, 2010.

[29] Nangate FreePDK45-nm Library, http://www.si2.org, 2011.