Low-Energy Encryption for Medical Devices: Security Adds an Extra Design Dimension

Junfeng Fan, Oscar Reparaz, Vladimir Rožić, Ingrid Verbauwhede
KU Leuven - COSIC and IMINDS
Kasteelpark Arenberg 10
Leuven, Belgium
{junfeng.fan,oscar.reparaz,vladimir.rozic,ingrid.verbauwhede}@esat.kuleuven.be

ABSTRACT

Smart medical devices will only be smart if they also include technology to provide security and privacy. In practice this means the inclusion of cryptographic algorithms of sufficient cryptographic strength. For battery operated devices or for passively powered devices, these cryptographic algorithms need highly efficient, low power, low energy realizations. Moreover, unique to cryptographic implementations is that they also need protection against physical tampering either active or passive. This means that countermeasures need to be included during the design process.

Similar to design for low energy, design for physical protection needs to be addressed at *all design abstraction levels*. Differently, while skipping one optimization step in a design for low energy or low power, merely reduces the battery life time, skipping a countermeasure, means opening the door for a possible attack. Designing for security requires a thorough threat analysis and a balanced selection of countermeasures.

This paper will discuss the different abstraction layers and design methods applied to obtain low power/low energy and at the same time side-channel and fault attack resistant cryptographic implementations. To provide a variety of security features, including location privacy, it is clear that medical devices need public key cryptography (PKC). It will be illustrated with the design of a low energy elliptic curve based public key programmable co-processor. It only needs $5.1\mu J$ of energy in a 0.13 μm CMOS technology for one point multiplication and includes a selected set of countermeasures against physical attacks.

1. INTRODUCTION

It is widely known that medical data needs the highest protection against disclosure and against tampering [2]. Indeed medical devices and medical data have a long lifespan. For instance, the battery of a pacemaker will last for 5 to 15 years before it is replaced. In addition, pacemakers and other medical (implantable) devices, wireless sensors, RFID tags, and others have become more sophisticated over the years. Instead of only issuing a fixed electrical pulse, they are now tuned to the patient. Similar arguments can be made for body sensors based on BAN, WAN or RFID technology: they pick up vital signs and transmit them to a wearable collector of data, e.g. the patient's cellular phone. The longevity of medical data explains the need for security levels that last for many years , since the attackers only get stronger over time due to Mooer's law. Unfortunately, longer key length translates in a larger computational load.

On top, the devices itself are not protected inside computer rooms or behind walls. Therefore, physical attacks, active or passive, are possible. For instance, pacemakers can be remotely updated or tuned. This wireless link can be eavesdropped, or it can be used to interfere with the readings or settings of the pacemaker. Wireless tags which are used to monitor the health status, give a patient much more freedom of movement and allow medical staff to monitor a patient without being bedridden. However, this can also be used to track patients and therefore location privacy is an important concern.

The goal of this paper is to provide insight on how to combine efficiency (in terms of energy or power consumption) with high security levels. We claim that this can only be reached by considering all design abstract levels. In this sense, design for security is similar to design for low power or low energy. It is also different, as a designer has to decide which is the right abstraction level to address particular attacks.

To illustrate this, the paper is organized as follows. We first describe some typical scenario's and the associated security analysis in section 2. A security analysis is used to select the type of cryptographic algorithms and protocols required for the application. In section 3, we discuss the design abstraction levels, which we call the security pyramid. Algorithms and protocols to address the security requirements are addressed in section 4. The architecture level is discussed in section 5 and the circuit level in section 6. Finally, the security evaluation is discussed in section 7.

2. SECURITY ANALYSIS

In a typical scenario, we assume that a patient wears several medical devices and sensors, some of them are worn on the body such as a hearing aid, or an insulin pump, others are implanted, such as a pacemaker or a brain monitoring or stimulation device [2].

In this typical scenario, these sensors and actuators communicate over a wireless channel: this could be a BAN

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

(body-area-network), RFID, Zigbee or similar. One example is the Human++ project of IMEC [15]. In this scenario a device, a cellular phone or similar, will serve as the local mini server, meaning that it collects the data and controls the network. It is assumed that this mini server is energy rich compared to the sensors and actuators. In a typical use case, the sensors will transmit patient data, e.g. his or her vital signs to the mini-server. Confidentiality, source authentication as well as integrity of this data are important. This is needed in both directions, i.e. mutual authentication is needed.

Recently, privacy received a lot of attention, more specifically location privacy or resistance to tracking. While early protocols did not make a distinction in types of privacy [9], more recent papers aim at providing strong privacy levels [14].

A careful choice of the protocols is only half of the security analysis. It is a necessary but not sufficient requirement to attain the desired security goals such as confidentiality, authentication and integrity.

For the system to be secure, we also require the implementation of the protocols be secure against an adversary that may have physical or short-distance access to the medical device. Techniques belonging to the broad class of side-channel attacks, such as Power Analysis [7, 8] have been used to extract keys from embedded devices by only monitoring the execution time of a cryptographic computation or the power consumption of the device (even in a remote contact-less fashion with specialized antennas that pick up the electromagnetic emanations of the chip).

In general, side-channel attacks exploit additional information that is available during the cryptographic computation aside from the input and output values, in contrast to classical cryptanalytical attacks. It is clear that should an attacker extract cryptographic keys from an embedded medical device, the security of the protocol is compromised.

Hence, the implementation of a medical embedded device should provide some degree of protection against physical attacks, such as tampering or side-channel attacks. In the next sections, we address the design procedure, best practices and decisions that are made in different abstraction levels when designing an exemplary crypto-processor for efficiency, low power, low energy and security.

3. DESIGN METHODS FOR LOW POWER AND SECURITY

Over the years, many design methods for low power and/or low energy have been developed. These design methods are situated at different abstraction levels. It is generally accepted and shown in practice that optimizations at higher abstraction levels have a bigger impact than those at lower abstraction levels.

An early paper showing the importance of transformations for low power is the paper by Chandrakasan et al. [3]. It shows that techniques like pipelining and parallelism can be used to reduce the power consumption. Others showed the importance of transformations or techniques at system level, e.g. the memory transformations introduced in [13]. At circuit level, also many techniques have been introduced: well known examples are gated clock strategies, reduced swing strategies or the introduction of power domains.

We distinguish the following abstraction levels for our purpose [21], as shown in Figure 3. The top level is the application or 'system' level. The selection of the protocol and the associated cryptographic algorithms has a huge influence on the final power or energy consumption of the embedded medical device. Next, is the algorithm level. For secret key and certainly for public-key algorithms there is a wide selection of algorithms and implementation strategies. The next level is the architecture level, where a digital platform for the implementation of the protocol and algorithm is selected. In most embedded applications that require extra physical protection, a HW/SW co-design platform is chosen. Typically there is an embedded micro-controller with programmable co-processors that support cryptographic algorithms. This is typically the platform of choice for applications like RFID tags, smart cards, portable devices and also medical devices. The design is eventually mapped to circuit-level implementation. In order to resist side-channel analysis, circuit-level countermeasures are crucial.

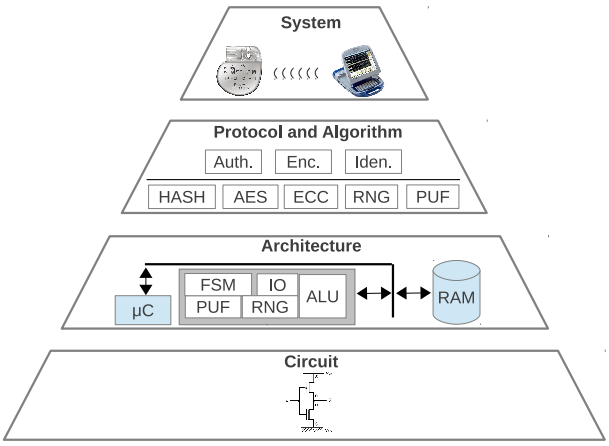

Figure 1: Security pyramid.

4. PROTOCOL AND ALGORITHM

Protocols are designed based on a variety of cryptographic primitives, such as hash functions, symmetric key ciphers, public key ciphers. It can also include non-algorithmic primitives, such as Random Number Generators (RNG), secure storage, or Physically Unclonable Functions (PUFs). Traditionally, protocol designers consider the security of a protocol the first design priority, with minimizing the computational complexity the second. Since implanted medical devices has strict power and energy budget, protocol designers need to consider many factors besides security. We identify a few of them below.

- *Security properties.* The security properties of a protocol should be clearly specified. In case of a pacemaker, mutual authentication is required to prevent impersonation. In order to protect the privacy of the patient, sensitive data should also be encrypted. Note that a modification on the ciphertext may also lead to a corrupted therapy that endanger the patient's life. Therefore, data authentication is also required. As such, the communication protocol between the pacemaker and the server should at least include mutual authentication, data authentication and encryption.

- *Location privacy* is a separate concern. It protects the user against tracking. Location based services are offered as part of many phone applications. In this case, accepting or denying it, should be the users choice. In medical applications, e.g. tracking as a means of protecting older people, should be strictly limited to medical personal. Location privacy heavily relies on public key based protocols.

- *The asymmetry between the parties* in a protocol, for instance a tag and a reader should also be considered in the protocol design. Protocols should be designed such that the heaviest computation load is for the reader (or other energy rich device) while the load for a tag or a sensor is minimized. This reduces the computation load. Other options are specific for the interaction of light-weight internet of things devices and are based on threshold cryptography [18].

- *Implementation size.* Implanted medical devices have a strict budget for the cryptographic modules: silicon area for hardware implementations or code size for software implementations. Close interaction with implementation people is needed: e.g. protocol designers tend to believe that hash functions are very cheap in hardware, thus should be used in light-weight protocols. For the most recent generation of hash functions, this is no longer true. The smallest SHA-1 implementation [12] uses 5527 gates, while an ECC core uses about 12k gates [10].

- *Energy usage reduction.* Protocols can be improved in at least three ways to reduce the energy usage on the device. Firstly, the computation on the device should be reduced as much as possible. Secondly, the communication should be minimized since wireless communication is power-hungry. Thirdly, the protocol should be designed to minimize energy consumption due to *useless* computations. Consider the mutual authentication between a pacemaker and a server, server authentication should be performed before other operations. As such, the protocol session stops immediately on the device when the server authentication fails.

Protocols based on secret key algorithms, like AES, are often cheaper in computation cost but not necessarily in communication cost. Secret key algorithms have also the problem of key distribution and management. Several exercises to evaluate the computation versus communication cost of secret-key versus public-key based security protocols have been made: the conclusions depend on the cryptographic algorithm, the digital platform and the wireless distance over which the communication occurs [5, 4].

Vaudenay [20] showed that public key algorithms are needed in order to provide strong privacy. However, not all PKC-based protocols achieves strong privacy. For example, tags using the Schnorr identification protocol [17] can be easily traced. We use the identification protocol by Peeters and Hermans [14] as an example. The protocol is shown in Figure 4, and it achieves *wide-forward-insider* privacy [1].

[1] *Wide-forward-insider* privacy is a widely used privacy notion in security analysis of private RFID identification protocols. It covers most practical use cases of private RFID identification. For a detailed definition, see [14].

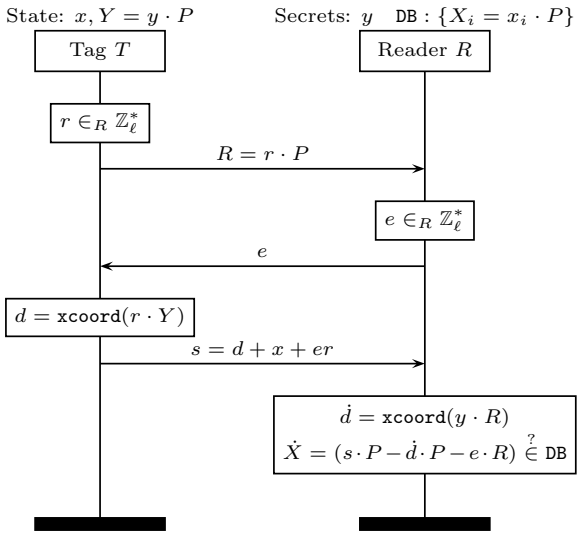

Figure 2: Peeters-Hermans identification protocol [14].

Elliptic curve cryptography is a favorable choice due to its relatively small key size and high security level compared to RSA. An elliptic curve over a finite field $\mathbb{F}_{2^m}$ is defined using the simplified Weierstrass function:

$$y^2 + xy = x^3 + ax^2 + b, \tag{1}$$

where $a, b \in \mathbb{F}_{2^m}$ are curve parameters. All the points $P(x, y)$ on the curve together with a point at infinity form an Abelian group $E(\mathbb{F}_{2^m})$. We can also add two points and obtain another point on the curve. Given an integer k, $k <$ Order(P), we can define $Q = k \cdot P = P + P + ... + P$ (k times). The computation of $k \cdot P$ is called elliptic curve point multiplication (ECPM). The security of ECC is based on the so-called Elliptic Curve Discrete Logarithm Problem (ECDLP), namely, given Q and P to find k such that $Q = k \cdot P$. ECDLP is believed to be computationally infeasible.

As shown in Figure 4 the main operation on the tag is two point multiplications (namely, $r \cdot P$ and $r \cdot Y$), and one modular multiplication (namely, er). All the operations should be protected against side-channel attacks and fault attacks. The challenge is how to securely perform these operations and at the same time meet the area and power budget.

The first step is to select curve parameters, which largely determines the security level and implementation size. Our ECC chip uses a Koblitz curve [1] defined over $\mathbb{F}_{2^{163}}$, which provides 80-bit security, equivalent to 1024-bit RSA. Besides, multiplication in binary extension fields it is carry-free. As a result, the multiplier is smaller and faster than integer multipliers.

The point multiplication algorithm directly determines the performance, the size of temporary storage, the performance and also its resistance against side-channel attacks. Our ECC chip uses the Montgomery powering ladder (MPL) for ECPMs. Note that MPL also allows us to use only the x coordinate to represent a point. One coordinate requires 163 bits of memory. Our ECC chip uses six 163-bit registers for the whole point multiplication. On the contrary, the best known algorithm for ECPM over a prime field uses

Algorithm 1 Point Multiplication using MPL

Require: An EC $y^2+xy = x^3+ax^2+b$, a point $P = (x, y)$, a t-bit integer k, $k = (1, k_{t-2}, ..., k_0)$, $k_i \in \{0, 1\}$
Ensure: $R = kP$
  $R \leftarrow (xr, r)$ //projective coordinate randomizaton
  $Q \leftarrow 2 \cdot P$
  for $i = t - 2$ downto 0 **do**
    if $k = 1$ **then**
      $R \leftarrow R + Q$, $Q \leftarrow 2 \cdot Q$
    else
      $Q \leftarrow Q + R$, $R \leftarrow 2 \cdot R$
    end if
  end for
  $R \leftarrow RecoverY(P, R)$
  Return R

8 registers excluding a and b [6]. The MPL algorithm is also resistant against Timing and Simple Power Analysis Attacks. In order to prevent Differential Power Analysis, we use randomized projective coordinates. More details about the countermeasures are discussed in the following section.

5. ARCHITECTURE LEVEL

At this level, we still have the same design dimensions (area, speed, power, energy, security) but estimations become more accurate. Optimizing the design on one dimension may lead to deterioration on the others. Here we describe several methods for algorithm level optimization.

- *Identify the root of trust.* Adding countermeasures leads to larger area or longer running time. Therefore, it is important to partition the design into a *secure* zone and an insecure zone. The secure zone operates on the sensitive data, this part should be protected using state-of-the-art countermeasures. The insecure zone contains the non-critical parts of the systems such as parts of the algorithm that don't depend on the secret information: this part can be implemented using a standard design flow. As long as the insecure zone is not compromised, the security of the system as a whole remains. One elegant solution is using a secure co-processor for the critical parts of the algorithm and an ordinary processor for everything else. The secure co-processor can then be strengthened by applying the countermeasures at the circuit level, or even using a full-custom approach.

- *Architecture-level security evaluation.* A crypto co-processor usually includes both hardware and software. The hardware part helps to achieve a high energy efficiency, while the software part provides flexibility. Sensitive data should appear only on the internal data-bus, and should not be available through the instruction set. So, no strange combination of instructions should release the key or the private date. For example, a procedure that reads the secret key from the memory and sends it to the output should not be programmable with the given instructions. Moreover, countermeasures against side-channel attacks need to be included. At a minimum, to prevent timing attacks, all instructions should execute with a constant number of cycles.

- *Area-power-security trade-off.* Although global optimization seems to be difficult, local optimization is possible for the trade-off between area, execution time and power consumption. For instance, in our ECC co-processor, a digit-serial multiplier for $\mathbb{F}_{2^{163}}$ is used. The choice of the digit-size determines the power needed for the computation, as well as the latency and area [16]. By using a digit serial multiplication with a 163×4 modular multiplier we achieve the optimal area-energy product within the given latency constraints. Moreover, the execution time is independent of the key length.

6. CIRCUIT LEVEL

If the basic design building blocks, the logic gates, are not designed to support security, the problem propagates to higher levels of abstraction and compromises the security of the complete system. Ignoring the problems at the circuit level, can make countermeasures at protocol, algorithm or architecture level irrelevant.

Most digital integrated circuits use standard CMOS logic due to its compact area, low power and availability of a standard cell design methodology. However, CMOS circuits have one fundamental security weakness. During the $0 \rightarrow 1$ transition at the output, a CMOS gate consumes power from the source which is not the case for $0 \rightarrow 0$, $1 \rightarrow 1$ or $1 \rightarrow 0$ transitions. This asymmetry is what enables the attacker to develop a power consumption model and, by comparing the model prediction with the actual measurements, extract the secret information.

Sense amplifier based logic (SABL) consumes the same amount of energy regardless of the data being processed which is achieved by using complementary outputs and dynamic operation. In order to have a meaningful improvement, this countermeasure has to be accompanied by a balanced layout of dual signal wires. Alternative style, Wave Dynamic Differential Logic (WDDL) operates using the same principle, and is compatible with regular synthesis, and place and route tools [19]. Side-channel resistant logic styles are the most efficient countermeasures to prevent power analysis, however they come with high area and power cost.

Making the power consumption data independent seems to be the most promising approach so far. Even when no dedicated logic style is used, there are several tricks that can be used to reduce the information leakage in combination with standard cell based design. These tricks do not provide the same level of protection as using specialized logic styles do, but they still increase the attack effort in practice.

- *Balance critical signals* to reduce the risk of SPA. Critical signals are typically control signals driving the multiplexers. These control signals usually connect to many multiplexers (164 in the presented ECC co-processor) as well as to a complex network of long wires and signal repeaters. Due to this high capacitive load, signal transitions will cause a noticeable pattern in the power trace. E.g. Figure 6 illustrates how multiplexer control signals can depend on the value of the key bit k. These signals have to be encoded in such way that the corresponding hamming differences are constant, otherwise the unbalance will reflect in the power trace. Regular layout structure and identical routing of these signals will make this countermeasure more effective.

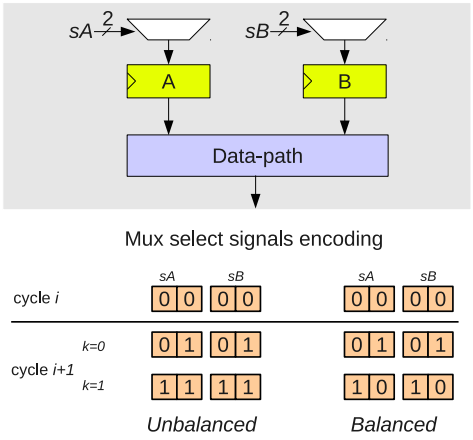

Figure 3: Register updating scheme and multiplexer encoding.

- *Avoid data-dependent clock-gating.* Clock gating may be a tempting solution to reduce dynamic power, however, in some cases, overly aggressive clock gating can introduce security risks. If different registers are enabled depending on the secret key different parts of the clock tree will be activated. The corresponding difference in power consumption will result in a clearly visible pattern in the power trace, thereby enabling an SPA. The mere fact that a different set of registers is gated, can be linked to a particular instruction sequence and directly or indirectly to the key.

- *Isolate the inputs to the data-paths.* When register outputs are connected to the data-path, updating the register value will cause spurious signal transitions inside the data-path. This will increase the power consumption but it will also compromise the security since the power consumption is correlated with the data loaded to the register. The solution is to set data-path inputs to a fixed value when it is not used. This can usually be implemented using AND gates and enable signals.

- *Avoid glitches.* This is a good practice for low power design since unwanted glitches result in higher power consumption. Even when the number of $0 \rightarrow 1$ transitions is balanced at the higher abstraction levels, glitches that appear in the data-path can cause data-dependent power consumption thereby enabling an attack. Please note that dynamic differential logic (such as SABL and WDDL) provide inherent protection against glitching. Other circuit styles were broken based on glitches [11].

The presented circuit-level optimization techniques reduce the risk of side-channel attacks and increase the security of the system. Some of these design practices align well with the power reduction techniques, while others are in clear contradiction.

A prototype chip for the ECC co-processor, based on the architecture presented in [10] is fabricated using UMC $0.13\mu m$ process. At the operating frequency of $847.5kHz$ and core voltage $Vdd = 1V$, the processor consumes $50.4\mu W$ and uses only $5.1\mu J$ for one point-multiplication. At this frequency, the throughput is 9.8 point multiplications per second.

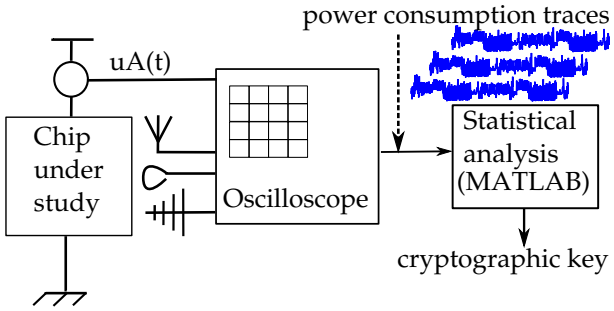

Figure 4: Typical workflow for side-channel attacks

7. SECURITY EVALUATION

A security evaluation typically starts with a white-box evaluation of a prototype chip and system. In a white-box evaluation, the attacker has complete access to the inner working details of the chip, including a precise description of the countermeasures implemented, and is generally regarded as a worst-case evaluation. A real attacker, later-on in a practical setting, will have less information available. The countermeasures used in the prototype co-processor were evaluated in a worst-case lab setting as Figure 4 depicts. The setup allows high sampling resolution of the instantaneous power consumption of the device.

Timing attacks exploit the timing variance with different inputs to provide some information about the key [7]. The prototype co-processor is intrinsically resistant to timing attacks. This is due to the fact that the computation time of a point multiplication is the same for different key values. This is achieved by careful optimizations on two abstraction levels. At the algorithm level, the Montgomery powering ladder requires the same number of iterations, while at architecture level, it is ensured that each iteration uses a constant number of clock cycles.

Moreover, since the same operations are executed in the same order in every invocation of the scalar multiplication routine (regardless of the value of the key), the device is mostly secure against attacks that inspect the power consumption signature of the device, a.k.a Simple Power Analysis (SPA) attacks. We identified a complex attack that could extract the key since a small source of SPA leakage was detected in our white-box evaluation. However, in order for the attacker to exploit it, he has to perform a complex profiling phase with an identical device that is under his total control, which is outside of the scope of our initial requirements. One of the causes of this SPA leakage might be that, although at the layout level the design was carefully balanced, slight unbalances are still present in the layout.

Differential Power Analysis (DPA) [8] is a statistical technique used to recover the key of a cryptographic chip from its instantaneous power consumption, provided that the power consumption is related to the intermediate data processed. Informally, DPA recovers the key in a divide-and-conquer fashion by comparing the measured power consumption with several hypothesized power consumptions, one for each subkey hypothesis. It is generally expected that the similarity between the measured power consumption and the predicted

power consumption will be high only for the correct key hypothesis. Note that the power consumption can be picked up remotely from the electromagnetic emission of the chip by using specific-purpose antennas.

To prevent DPA, the chip randomizes the internal points representation by using a random Z coordinate in each execution. Since the intermediate values cannot be predicted, DPA attacks cannot be mounted.

We empirically confirmed that DPA attacks are correctly thwarted by using randomized projective coordinates. When the countermeasure is disabled, a DPA attack succeeds with as low as 200 traces. When the countermeasure is enabled, but the randomness is known, the attack also succeeds. This scenario is only possible in a white-box evaluation and does not correspond to the normal operation of the chip. In the normal operation, the randomness is generated by the chip and kept secret to the adversary. The fact that the attack works in this lab setting provides confidence on the soundness of the attack. When the countermeasure is enabled, and the randomness is unknown, the attack does not succeed. Even 20000 traces are not enough to reveal a single key bit, using the same DPA attack.

8. CONCLUSIONS

Making a device secure adds an extra design dimension. Indeed, for the design of medical devices, a trade-off between security, power and energy needs to be made. We have described the security traps on each abstraction level and presented the corresponding design guidelines. This is illustrated with the design of a light-weight co-processor for elliptic curve cryptography.

9. ACKNOWLEDGMENTS

Oscar Reparaz is funded by an FWO fellowship. In addition, this work is supported in part by the Flemish Government through FWO G.0550.12N and G.0130.13, the Hercules Foundation AKUL/11/19, and by the European Commission through the ICT program under contract FP7-ICT-2011-284833 PUFFIN.

10. REFERENCES

[1] FIPS PUB 186-3, Digital Signature Standard (DSS).

[2] W. P. Burleson, S. S. Clark, B. Ransford, and K. Fu. Design Challenges for Secure Implantable Medical Devices. In *DAC 2012*, June 2012. Invited paper.

[3] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. Brodersen. Optimizing Power Using Transformations. *IEEE TCAD*, 14(1):12–31, 1995.

[4] G. de Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira. On the Energy Cost of Communications and Cryptography in Wireless Sensor Networks. In *(extended version), SecPriWiMob 2008*, pages 580–585, 2008.

[5] A. Hodjat and I. Verbauwhede. The Energy Cost of Secrets in ad-hoc Networks. In *Proc. IEEE Circuits and Systems Workshop on Wireless Communications and Networking*, page 4, 2002.

[6] M. Hutter, M. Joye, and Y. Sierra. Memory-Constrained Implementations of Elliptic Curve Cryptography in Co-Z Coordinate Representation. In *AFRICACRYPT*, pages 170–187, 2011.

[7] P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology - CRYPTO*, pages 104–113, 1996.

[8] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology - CRYPTO*, pages 388–397, 1999.

[9] Y. Lee, L. Batina, and I. Verbauwhede. EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID Authentication Protocol. In *IEEE International Conference on RFID*, pages 97–104, 2008.

[10] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede. Elliptic-Curve-Based Security Processor for RFID. *IEEE Trans. Computers*, 57(11):1514–1527, 2008.

[11] S. Mangard, T. Popp and B. M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In *CT-RSA*, pages 351–365, 2005.

[12] M. O'Neill. Low-cost SHA-1 Hash Function Architecture for RFID Tags. In *Workshop on RFID Security - RFIDSec*, pages 41–51, 2008.

[13] P. R. Panda, F. Catthoor, N. D. Dutt, K. Danckaert, E. Brockmeyer, C. Kulkarni, A. Vandecappelle, and P. G. Kjeldsberg. Data and Memory Optimization Techniques for Embedded Systems. *ACM Trans. Design Autom. Electr. Syst.*, 6(2):149–206, 2001.

[14] R. Peeters and J. Hermans. Wide Strong Private RFID Identification Based on Zero-Knowledge. *IACR Cryptology ePrint Archive*, 2012:389, 2012.

[15] V. Pop, R. de Francisco, H. Pflug, J. Santana, H. Visser, R. J. M. Vullers, H. de Groot, and B. Gyselinckx. Human++: Wireless Autonomous Sensor Technology for Body Area Networks. In *ASP-DAC 2011*, pages 561–566, 2011.

[16] K. Sakiyama, L. Batina, B. Preneel, and I. Verbauwhede. Multicore Curve-Based Cryptoprocessor with Reconfigurable Modular Arithmetic Logic Units over GF(2^n). *IEEE Trans. Computers*, 56(9):1269–1282, 2007.

[17] C.-P. Schnorr. Efficient Identification and Signatures for Smart Cards. In G. Brassard, editor, *Advances in Cryptology - CRYPTO'89, LNCS*, volume 435, pages 239–252. Springer-Verlag, 1989.

[18] K. Simoens, R. Peeters, and B. Preneel. Increased Resilience in Threshold Cryptography: Sharing a Secret with Devices That Cannot Store Shares. In *Pairing 2010*, volume 6487 of *LNCS*, pages 116–135, 2010.

[19] K. Tiri and I. Verbauwhede. A Digital Design Flow for Secure Integrated Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(7):1197–1208, 2006.

[20] S. Vaudenay. On Privacy Models for RFID. In *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 68–87, 2007.

[21] I. Verbauwhede and P. Schaumont. Design Methods for Security and Trust. In *DATE 2007*, pages 1–6, NICE, FR, 2007. IEEE.