Anti-counterfeiting, Key Distribution, and Key Storage in an Ambient World via Physical Unclonable Functions*

Jorge Guajardo[1], Boris Škorić[1], Pim Tuyls[1], Sandeep S. Kumar[1], Thijs Bel[1],
Antoon H. M. Blom[2], and Geert-Jan Schrijen[1]

[1] Philips Research Europe, Eindhoven, The Netherlands
{jorge.guajardo,boris.skoric,pim.tuyls}@philips.com
{sandeep.kumar,thijs.bel,geert.jan.schrijen}@philips.com

[2] Philips Applied Technologies, Eindhoven, The Netherlands
a.h.m.blom@philips.com

Abstract. Virtually all applications which provide or require a security service need a secret key. In an ambient world, where (potentially) sensitive information is continually being gathered about us, it is critical that those keys be both securely deployed and safeguarded from compromise. In this paper, we provide solutions for secure key deployment and storage of keys in sensor networks and RFID systems based on the use of Physical Unclonable Functions (PUFs). In addition, to providing an overview of different existing PUF realizations, we introduce a PUF realization aimed at ultra-low cost applications. We then show how the properties of Fuzzy Extractors or Helper Data algorithms can be used to securely deploy secret keys to a low cost wireless node. Our protocols are more efficient (round complexity) and allow for lower costs compared to previously proposed ones. We also provide an overview of PUF applications aimed at solving the counterfeiting of goods and devices.

Key Words. Physical Unclonable Functions, Intrinsic PUF, SRAMs, LC-PUFs, Fuzzy Extractor, Helper Data Algorithm, Sensor Nodes, Key Distribution.

1 Introduction

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." This is how Mark Weiser began the paper (Weiser, 1991), where he introduces for the first time the term Ubiquitous Computing (Ubicomp) and where the field of ubiquitous computing was born. The basic idea of Ubicomp is that computers (understood in a broad sense) will vanish into the background of our lives, to such an extent that we will interact every day with them without even noticing. Today, the world is one step closer to Weiser's vision thanks in large part to technologies such as sensors, Radio Frequency Identification (RFID), PDAs, cellular phones, etc.

Apart from their many benefits, the ubiquity of such devices creates security and privacy problems that would not exist otherwise. We will focus on the particular case of sensors and RFID[3] as these are some of the most invisible technologies present today, enabling the ubicomp vision. Sensors, for example, are expected to be embedded and distributed everywhere becoming the eyes and ears of the world around us. They will allow us to interact with our environment (and vice versa) in a transparent manner. Sensors also have the potential of being used in areas as diverse as medical applications (Shnayder et al., 2005), emergency response (Lorincz et al., 2004),

[*] Appears in Information Systems Frontiers, Special Issue on Ambient Intelligence, Online Publication: October 23, 2008. ©Springer. DOI: 10.1007/s10796-008-9142-z

[3] We use the term RFID in this paper in very broad sense. In particular, we consider an RFID tag as any device that communicates in the radio frequency range of the spectrum and that can be used for identification purposes. For a taxonomy of RFID tags see for example Sarma and Engels (2003); Engels and Sarma (2005).

monitoring volcanic activity (Werner-Allen et al., 2006), information aggregation applications such as real time traffic monitoring, wildfire tracking, wildlife monitoring, or building safety monitoring (Przydatek et al., 2003), smart dust applications (Hsu et al., 1998), and many others. A common characteristic of all these applications is the gathering of data (after all, that is the job of a sensor node) which could be security or privacy sensitive. Thus, it is no surprise that both the academic and industry communities have shown a lot of interest in developing security solutions (protocols, primitives, etc.) to respond to (potential) security and privacy problems. We refer to Chan and Perrig (2003); Perrig et al. (2004) for a survey of security problems and solutions in sensor networks.

The RFID case is similarly interesting. RFID as a technology is a rather old one, dating back to the Second World War when the Royal Air-force used it to identify allied planes from their enemy counterparts (Landt, 2001; Eagle, 2002). However, it was not really until 1999, when RFID started to experience a boom. This boom's main reason was, and continues to be, the envisioned ubiquity of RFID tags in everyday life. In fact, RFID tags are expected to be embedded in (or associated with) every object we come in contact with: from clothes to posters, from microwaves to food packages, from the smallest to the largest, thus enabling the so-called Internet of Things. The pervasiveness of RFID tags, their ability to carry more information than bar codes, their expected low cost (below 10 US dollar cents), and their lack of need for line of sight communication also pose interesting challenges to those interested in their widespread adoption. Such challenges include both privacy and security concerns. On the privacy front, we can identify concerns on the part of consumers who will be carrying tagged objects. In particular, the wireless communication capabilities of RFID tags and their simple functionality (when queried they simply reply with their unique identifier) make it possible to track people based on tag identifiers as well as to find out consumer preferences clandestinely. Similarly, companies and defense organizations will also be more vulnerable to espionage as it will be much easier to gather information about the competition or the enemy and much harder to detect such spying activities. We refer the reader to Juels et al. (2005); Juels (2006) for a comprehensive survey of privacy issues in RFID.

On the security front, we have the authentication problem. In other words, how a legitimate party can assess whether an RFID tag associated with an object and the object itself are authentic. The ability to authenticate legitimate tags has direct implications on industry's ability to reduce the counterfeit market, which in 2004 was estimated to surpass the 500 billion USD per year mark (ICC; Staake et al., 2005). The counterfeiting problem has been shown to be a significant threat to both enterprises and individuals as the following examples show: (i) in 2005, Bono et al. (2005) showed how a popular transponder built by Texas Instruments and used by several automobile manufacturers in their ignition keys could be successfully cloned and (ii) Carluccio et al. (2006 a,b) show how to build cheap RFID readers which could be used to trace individuals via RFID chips embedded in passports. Clearly, the damage that counterfeited products generate is not limited to tangible losses in terms of revenues but also includes a damaged brand and reputation as well as human death in extreme cases (see e.g. Lacey 2006).

Many solutions have been developed for the previously mentioned privacy and security problems (see e.g. Chan and Perrig 2003; Perrig et al. 2004; Wong et al. 2004; Juels et al. 2005; Juels 2006; Guajardo et al. 2008b to get overviews of both areas). It is natural that all security and privacy preserving protocols use some sort of secret-key material regardless of whether the protocols are based on public-key or private-key cryptography. The interesting fact, however, is that everyone assumes that the key is magically deployed onto the nodes of the network in a safe and secure manner as noted most recently by Kuo et al. (2007). Most notably, one of the best examples of sensor node deployment in the "real world," the Zigbee Specification (Zig, 2005), assumes that either the nodes will be loaded with their key material by sending the keys in the clear (resulting in a brief vulnerability window) or that factory initialized keys are pre-loaded on the nodes. Kuo et al. (2007) notice, however, that such factory pre-set keys might not be trusted by many users.

A second interesting development (mostly) relevant for RFID applications is that whereas there has been a lot of work done on secure protocols, very few people have considered the physical security of the actual tags. After all, one of the simplest attack that one can imagine on such cheap devices is to tamper with them and extract their secrets, either reading their memory contents or performing a physical attack on the tag (see e.g. Oren and Shamir 2006). Thus, it would be interesting to provide tamper resistance for such cheap devices[4]. Notice that tamper resistance also provides forward secrecy, in the sense that if it is much harder to compromise the key of a single node, then the keys of previously deployed nodes are also safeguarded.

1.1 Our contributions

In this paper, we propose a new method for secure key deployment of sensor node keys based on the properties of Physical Unclonable Functions (PUFs) and fuzzy extractor schemes. The advantages range from the added security guarantees provided by tamper evidence, tamper resistance and unclonability as provided by PUFs, to significantly simplified protocols which make the life of the end-user (the individual deploying a wireless sensor network) easier. In addition, we show that under relaxed (but reasonable) security assumptions we can provide costs reduction, since our protocols do not require additional hardware set-up devices as the Message-In-a-Bottle (MIB) protocol (Kuo et al., 2007) does. Notice that we choose to compare to the MIB protocol since, to our knowledge, it is the only protocol that has thoroughly considered all requirements that must be satisfied by a key deployment protocol.

In addition, we describe several PUF physical realizations. We also provide a new PUF construction of independent interest, particularly suited to applications where the aim is to detect counterfeited products at a very low cost. Finally, we provide an overview of other PUF applications in the areas of anti-counterfeiting technologies, secure key storage, and authentication protocols.

1.2 Organization

In Section 3 we describe the idea of Physical Unclonable Functions (PUFs) and how we can use these noisy information sources as robust identifiers. In particular, we explain the idea of Fuzzy Extractors or Helper Data algorithms. Section 4 is devoted to describing known PUFs. We also describe a new PUF construction based on resonance peaks in the frequency response of randomized LC-circuits, where L and C refers to the inductance and capacitance present in the circuit, respectively. In Sect. 5, we introduce new protocols for the secure deployment of secret-key material in sensor nodes and analyze their advantages when compared to other protocols, particularly the work described in Kuo et al. (2007). Section 6 provides an overview of other PUF technology applications. We make particular emphasis on the anti-counterfeiting area given its relevance and impact on our everyday lives. We end with conclusions in Sect. 7.

2 Preliminaries

We briefly recall some definitions, which will be used in the remainder of the paper. Unless otherwise stated, we follow the presentation of (Dodis et al., 2004; Boyen, 2004).

[4] There are known tamper resistance methods to protect cryptographic material but, to our knowledge, none that would be economically viable for cheap applications such as RFID or sensor nodes.

Hamming Distance. The Hamming distance between two vectors $x, y \in Q^n$, where Q is some field is denoted by $\mathrm{dis}(x, y)$ and it is defined to be the number of coordinates in which they differ. For our applications Q will be a finite field of characteristic p and often of characteristic two.

Error Correcting Codes. A q-ary block code $\mathcal{C} = \{w_1, w_2, \ldots, w_k\}$ of length n is any non-empty subset of Q^n, where Q has cardinality q, i.e. Q has q distinct symbols. For example if Q is the Galois Field $\mathbb{F}_q$ then Q has q elements and q is a prime power. The elements w_i of $\mathcal{C} \subseteq Q^n$ are called the codewords. Notice that the w_is are n-tuples of symbols taken from the alphabet Q. The minimum distance of the code $\mathcal{C}$, written $d_{\min}$, is defined to be

$$d_{\min} := \min\{\mathrm{dis}(w_i, w_j) | w_i, w_j \in \mathcal{C}, w_j \neq w_i\}$$

For a given $d_{\min}$, the error correcting capability or error correcting distance e is:

$$e := \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

Geometrically it can be seen as the radius e such that for every element $w \in Q^n$ there is at most one codeword in the ball of radius e centered on w.

Linear Codes. A q-ary linear code $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q$. If $\mathcal{C}$ is a k-dimensional linear code of length n and minimum distance d, we write it as an $[n, k, d]$-code. Thus, a q-ary $[n, k, d]$-code has cardinality q^k, i.e., it can encode up to q^k possible messages. For linear codes the minimum distance is equal to the minimum non-zero weight in $\mathcal{C}$.

Permutation Groups. The set of all permutations of a set $\mathcal{M}$ is called the symmetric group on $\mathcal{M}$. Usually we take $\mathcal{M}$ to be the set $\{1, \ldots, n\}$, and denote the symmetric group by S_n, for some positive integer n. The order of S_n is $n!$. As it is well known, any permutation can be written as a product of disjoint cycles: we call this its cycle decomposition. For example, the permutation of $\{1, \ldots, 5\}$ which maps 1 to 4, 2 to 5, 3 to 1, 4 to 3, and 5 to 2 has cycle decomposition $(1, 4, 3)(2, 5)$. The cycle decomposition is unique up to writing the cycles in a different order and starting them at different points: for example, $(1, 4, 3)(2, 5) = (5, 2)(3, 1, 4)$. A permutation group $\mathcal{P}$ on a set $\mathcal{M}$ is a subgroup of the symmetric group on $\mathcal{M}$; that is, it is a set of permutations closed under composition and inversion and containing the identity permutation. The group operation is simply the action of the permutations π_i on the elements of the set $\mathcal{M}$. The permutation group $\mathcal{P} = \{\pi_i : \mathcal{M} \to \mathcal{M}\}$ indexed by i, is transitive on the set $\mathcal{M}$ if for any pair of points w, w' there exists a permutation $\pi_i \in \mathcal{P}$, such that $\pi_i[w] = w'$. The permutation group $\mathcal{P}$ is isometric with respect to the distance function dis in the set $\mathcal{M}$ (we assume the set $\mathcal{M}$ is a space with a distance function) if for all permutations $\pi_i \in \mathcal{P}$ and points $w, w' \in \mathcal{M}$, it holds that $\mathrm{dis}(\pi_i[w], \pi_i[w']) = \mathrm{dis}(w, w')$. These two last properties are used in the construction of fuzzy extractors based on permutations.

Universal Hash Functions (Carter and Wegman, 1979). A universal hash function is a map from a finite set $\mathcal{A}$ of size $|\mathcal{A}|$ to a finite set $\mathcal{B}$ of size $|\mathcal{B}|$. For a given hash function h and two strings x, x' with $x \neq x'$, we define the function $\delta_h(x, x')$ as equal to 1 if $h(x) = h(x')$ and 0 otherwise. For a finite set (or family) of hash functions $\mathcal{H}$, $\delta_{\mathcal{H}}(x, x')$ is defined to be $\sum_{h \in \mathcal{H}} \delta_h(x, x')$. In other words, $\delta_{\mathcal{H}}(x, x')$ counts the number of functions $h \in \mathcal{H}$ for which x and x' collide. For a random $h \in \mathcal{H}$ and any two distinct x, x', the probability that $h(x) = h(x')$ is $\delta_{\mathcal{H}}(x, x')/|\mathcal{H}|$, where $|\mathcal{H}|$ denotes the size of the set $\mathcal{H}$. There has been extensive research on universal hash functions (see for example Shoup 1996; Nevelsteen and Preneel 1999). In the hardware domain, their implementation has been investigated in Krawczyk (1994) and the work of Kaps et al. (2005).

3 Physical unclonable functions and helper data schemes

3.1 Overview

A function in mathematics is a relation which associates elements of a set $\mathcal{A}$, typically referred to as the domain, with elements of a set $\mathcal{B}$, known as the range or image. The relation which associates elements of set $\mathcal{A}$ to those of set $\mathcal{B}$ is defined via a mathematical formula, a graph, a table, etc. In 2001, Pappu (2001); Pappu et al. (2002) introduced the concept of Physical Random Functions or Physical Unclonable Functions. In this case the function is defined via a physical object or device. In particular, upon challenging such a PUF with a challenge C_i, a response R_i is generated. Thus, we write: $R_i \leftarrow \mathrm{PUF}(C_i)$. Physical Unclonable Functions have essentially two parts: i) a physical part and ii) an operational part. The physical part is a physical system that is very difficult to clone[5]. It inherits its unclonability from uncontrollable process variations during manufacturing. In the case of PUFs on an IC such process variations are typically deep-submicron variations such as doping variations in transistors. The operational part corresponds to the function. In order to turn the physical system into a *function* a set of challenges C_i (stimuli) has to be available to which the system responds with a set of sufficiently different responses R_i.

PUFs types and examples. We distinguish between two different classes of PUFs: strong and weak PUFs. First, a strong PUF accepts a large number of challenge response pairs (C_i, R_i), $i = 1, \ldots, N$; i.e. the PUF has so many CRPs such that an attack (performed during a limited amount of time) based on exhaustively measuring the CRPs only has a negligible probability of success and, in particular, $1/N \approx 2^{-k}$ for large $k \approx 100$ (Pappu, 2001; Škorić et al., 2005). If the number of different CRPs N is rather small, we refer to it as a weak PUF. Notice that a weak PUF is usually used for secure key storage applications and thus, it is very similar to the concept of Physically Obfuscated Keys (POKs) as introduced by Gassend (2003). Examples of PUFs include optical PUFs (Pappu, 2001; Pappu et al., 2002), silicon PUFs (Gassend et al., 2002b) and coating PUFs (Tuyls et al., 2006). In Guajardo et al. (2007b) the notion of an *Intrinsic* PUF (IPUF) was introduced. In other words, an IPUF is a PUF inherently present in a device due to its deep-submicron manufacturing process variations and no additional hardware has to be added for embedding the PUF. In Guajardo et al. (2007b), the authors show that the start-up values of SRAM memory cells (present, for example, in an FPGA) are an IPUF. In this paper, we also introduce weak PUFs based on measuring the resonance frequencies of LC-circuits. Such PUFs are particularly relevant in applications where low cost identification is important. We discuss in detail these and other types of PUFs in Sect. 4.

The need for helper data schemes. The responses of a PUF can not be used as a key (as in e.g. Tuyls et al. 2006) in a cryptographic primitive for two reasons. First, the responses of a PUF are obtained through measurements which are typically noisy. When a PUF is challenged with C_i, a response R_i' which is a noisy version of R_i is obtained. This leads to a problem since cryptographic functions are very sensitive to noise on their inputs. Second, the responses of a PUF are not uniformly distributed. Hence, even if there was no noise, the response would not form a cryptographically secure key. In order to deal with both issues a Fuzzy Extractor or Helper Data algorithm has to be used. For the precise definition of a Fuzzy Extractor and Helper Data algorithm we refer to Dodis et al. (2004); Linnartz and Tuyls (2003). This primitive deals with

[5] Note that this stands in sharp contrast to Quantum Cryptography where cloning is impossible due to the basic laws of nature. In the case of PUFs, there is a very small (but non-zero) probability that the structure can be cloned.

both issues by implementing first an *information reconciliation phase* and secondly, by applying a *privacy amplification* or randomness extraction primitive. We discuss fuzzy extractors in more detail in Sect. 3.3.

3.2 PUF security properties

As in any security system, in order to evaluate the security of the system, it is necessary that we state the necessary assumptions for the system to be secure. Previous works (Pappu, 2001; Gassend et al., 2002b; Tuyls et al., 2006; Guajardo et al., 2007a,b) have either explicitly or implicitly made the following assumptions:

1. It is assumed that a response R_i (to a challenge C_i) gives only a small amount of information on another response R_j (to a different challenge C_j) with $i \neq j$.
2. Without having the corresponding PUF (i.e. the actual physical device or structure) at hand, it is impossible to come up with the response R_i corresponding to a challenge C_i, except with negligible probability.

In most cases, it is also reasonable to assume that PUFs are tamper evident. This implies that when an attacker tries to investigate the PUF to obtain detailed information about its structure, the PUF is damaged. In other words, the PUF's challenge-response behavior is changed substantially.

As noticed previously, the above assumptions are guaranteed based on the hardness of copying the actual device (or structure) used as a PUF. This hardness is due to the infeasibility to copy the structure and it is not due to some physically impossible process. Thus, we can think of the unclonability property of PUFs as the physical equivalent of a computationally hard problem.

3.3 Fuzzy extractor and helper data schemes

One use of PUFs is as a source for cryptographic key material as noticed in Tuyls et al. (2006). Since PUF responses are noisy and the responses are not fully random, a Fuzzy Extractor or Helper Data algorithm is required to extract secure keys from the PUF responses. For formal definitions of Fuzzy Extractors and Helper Data algorithms we refer to Dodis et al. (2004); Linnartz and Tuyls (2003). Informally, we need to implement two basic primitives: (i) *Information Reconciliation* or error correction and (ii) *Privacy Amplification* or randomness extraction. In order to implement those two primitives, helper data W are generated during the *enrollment phase*. During this phase, carried out in a trusted environment, a probabilistic procedure called Gen is run. Later, during the *key reconstruction* or authentication phase, the key is reconstructed based on a noisy measurement R_i' and the helper data W. During this phase, a procedure called Rep is performed. We now present two constructions for such procedures previously described in Juels and Wattenberg (1999); Dodis et al. (2004). Constructions for other metrics can be found in Dodis et al. (2004).

Construction based on code offset. In order to implement the procedures Gen and Rep an error correction code $\mathcal{C}$ and a set $\mathcal{H}$ of universal hash functions (Carter and Wegman, 1979) is required. The parameters $[n, k, d]$ of the code $\mathcal{C}$ are determined by the length of the responses R and the number of errors t that have to be corrected. The distance d of the code is chosen such that t errors can be corrected.

The Gen-procedure takes as input a PUF response(s) R and produces as output a key K and helper data $W = (W_1, W_2)$. This is achieved as follows. First, a code word $C_S \leftarrow \mathcal{C}$ is chosen at random from $\mathcal{C}$. Then, a first helper data vector equal to $W_1 = C_S \oplus R$ is generated. Furthermore, a hash function h_i is chosen at random from $\mathcal{H}$ and the key K is defined as $K \leftarrow h_i(R)$. The helper data W_2 is set to i. Summarizing the procedure Gen is defined as follows, $(K; W) = (K; (W_1, W_2)) \leftarrow \mathsf{Gen}(R)$.

During the key reconstruction phase the procedure Rep is run. It takes as input a noisy response R' from the same PUF and helper data W and reconstructs the key K *i.e.* $K \leftarrow \mathsf{Rep}(R', W)$. This is accomplished according to the following steps: (1) *Information Reconciliation*: Using the helper data W_1, $W_1 \oplus R'$ is computed. Then, the decoding algorithm of $\mathcal{C}$ is used to obtain C_S. From C_S, R is reconstructed as $R = W_1 \oplus C_S$; and (2) *Privacy amplification*: The helper data W_2 is used to choose the correct hash function $h_i \in \mathcal{H}$ and to reconstruct the key as $K = h_i(R)$. Notice that we have implicitly assumed the use of a binary code. This construction is a variant of (Juels and Wattenberg, 1999) where the focus was on biometric applications.

Construction based on permutations. The permutation-based construction is due to Dodis et al. (2004). As in the code-offset construction, we choose a code $\mathcal{C} \subseteq \mathcal{M}$ and, in addition, a corresponding permutation group $\mathcal{P}$ that is both transitive and isometric. The $(K, W) \leftarrow \mathsf{Gen}(R)$ then computes K and W from input R by first selecting a random code word $C_S \leftarrow \mathcal{C}$ and corresponding $\pi_P \in \mathcal{P}$, such that $\pi_P[R] = C_S$. Notice that the transitivity property of $\mathcal{P}$ guarantees that such π_P will exist. Then as before, we randomly choose a universal hash function $h_i \in \mathcal{H}$ and we output $(K; (W_1, W_2)) = (h_i(R); (P, i)) \leftarrow \mathsf{Gen}(R)$.

During the key reconstruction phase a procedure called Rep is run according to the following steps: (1) *Information Reconciliation*: Using the helper data $W_1 = P$, we compute $\pi_P[R'] = C'_S$. Because of the isometric property of π, C'_S should be sufficiently close to C_S that, after applying the decoding algorithm of $\mathcal{C}$, we will obtain C_S. From C_S, R is reconstructed as $R = \pi_P^{-1}[C_S]$; and (2) *Privacy amplification*: The helper data W_2 is used to choose the correct hash function $h_i \in \mathcal{H}$ and to reconstruct the key as $K = h_i(R)$.

Security. The security of the above constructions has been established in Juels and Wattenberg (1999); Linnartz and Tuyls (2003); Dodis et al. (2004); Boyen (2004); Boyen et al. (2005). By security here we mean two complementary things. First, Linnartz and Tuyls (2003); Dodis et al. (2004) provide a bound on the number of bits of entropy left after the fuzzy extractor operates on the source bits of the PUF. In other words, given a number of bits with certain entropy, we know from Linnartz and Tuyls (2003); Dodis et al. (2004), how many "secure" bits we are left with after processing with the fuzzy extractor. Second, Juels and Wattenberg (1999); Boyen (2004); Boyen et al. (2005) show that given the public helper data information, negligible information is learned about the derived secret. Finally, Boyen (2004); Boyen et al. (2005) show how to protect the helper data against tampering and modification.

4 PUF realizations

This section describes four possible PUF realizations. The first three require specialized hardware designs. The last one is based on the start-up values of SRAM memories and, as a result, are often already present in devices such as FPGAs and microprocessors.

4.1 Optical PUFs and silicon PUFs

Pappu (2001); Pappu et al. (2002) introduced the idea of a Physical One-Way Function (POWF). They use a bubble-filled transparent epoxy wafer and shine a laser beam through it leading to a response interference pattern. This kind of analog PUF is hard to use in the field because of the difficulty to have a tamper resistant measuring device. Gassend et al. (2002a) define a Controlled Physical Random Function (CPUF) which can only be accessed via an algorithm that is physically bound to the randomness source in an inseparable way. This control algorithm can be used to measure the PUF and also to protect a "weak" PUF from external attacks. Škorić et al. (2007) also describe an CPUF based on an integrated optical PUF. Gassend et al. also introduce silicon Physical Random Functions (SPUF) (Gassend et al., 2002b) which use manufacturing process variations in ICs with identical masks to uniquely characterize each chip. The statistical delay variations of transistors and wires in the IC were used to create a parameterized self oscillating circuit to measure frequencies which characterize each IC. Silicon PUFs are very sensitive to environmental variations like temperature and voltage. Lim et al. (2005) introduce the concept of an *arbiter-based* PUF, which uses a differential structure - two identical delay paths - and an arbiter to distinguish the difference in the delay between the paths. Recently, Su et al. present in (Su et al., 2007) a custom built circuit array of cross-coupled NOR gate latches to uniquely identify an IC. Here, small transistor threshold voltage V_t differences that are caused by process variations lead to a mismatch in the latch to store a 1 or a 0.

4.2 Coating PUFs

In (Tuyls et al., 2006), Tuyls et al. present a coating PUF in which an IC is covered with a protective matrix coating, doped with random dielectric particles at random locations. The IC also has a top metal layer with an array of sensors used to measure the local capacitance of the coating matrix. These capacitance values are used to characterize the IC. The measurement circuit is integrated in the IC, making it a controlled PUF. Figure 1 shows a schematic diagram of the PUF construction. Figure 1 shows an schematic cross-section of the upper metal layer of an IC containing aluminum sensor structures (Al) that are used to measure the coating's local capacitance. Figure 2 shows a cross-section of an actual chip showing the coating, which includes random dielectric particles, and the sensor structures.

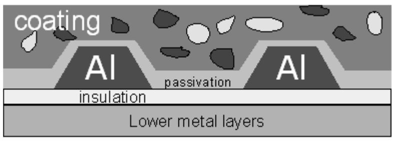


Fig. 1. Schematic cross-section of a Coating PUF IC.

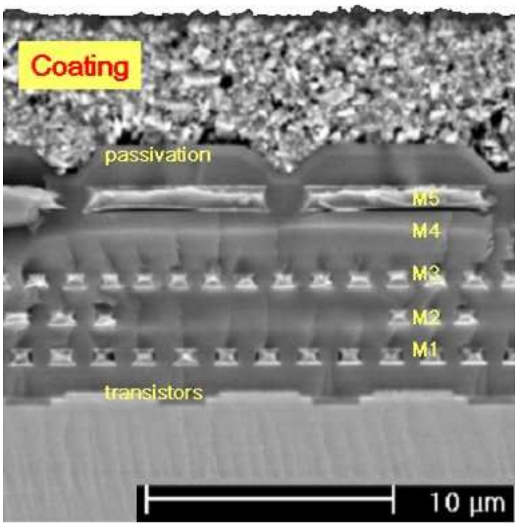


Fig. 2. Actual cross-section of a chip with security coating

It is shown in Tuyls et al. (2006) that it is possible to extract up to three key bits from each sensor in the IC. A key observation in Tuyls et al. (2006) is that the coating can be used to store keys (rather than as a challenge-response repository as in previous works) and that these keys are not stored in memory. Rather, whenever an application requires the key, the key is generated on the fly. This makes it much more difficult for an attacker to compromise secret-key material in security applications. Finally, Tuyls et al. (2006) show that active attacks on the coating can be easily detected, thus, making it a good countermeasure against probing attacks.

4.3 Random LC circuits as unclonable unique identifiers

A randomized capacitor, such as used in Coating PUFs (Tuyls et al., 2006), can also be employed more directly as a unique identifier that can be read out wirelessly. The capacitor is part of a completely passive resonator circuit ('LC circuit'). In addition to the capacitor, the circuit also comprises an inductor coil which serves as an antenna. The coil may have random properties as well. When a radio frequency (RF) electromagnetic field is generated in the vicinity of the antenna, the circuit absorbs an amount of power that depends on the frequency and on the precise characteristics of the capacitor and the coil. A frequency sweep yields a response curve that uniquely identifies the resonator circuit. If the LC circuit is difficult to clone we refer to it as an 'LC-PUF'. LC-PUFs are similar to the 'RF-COA' described in DeJean and Kirovski (2006). The main differences are that LC-PUFs are designed to have strong resonance peaks and that they do not require the same level of positioning accuracy.

Shape of the resonance signal. We consider the simplified case where all circuit components are 'ideal': a resistance R_1, a capacitance C_1 (with impedance $1/[i\omega C_1]$) and an inductance L_1 (with impedance $i\omega L_1$) connected in series. The impedance of this circuit is given by $Z_1(\omega) = R_1 + i\omega L_1 + 1/(i\omega C_1)$. The resonant frequency is $\omega_1 = 1/\sqrt{L_1 C_1}$. At this point $|Z_1|$ has its minimum. Similarly, the readout setup has R_0, C_0 and L_0. The coupling between the readout coil and the PUF is assumed to be purely inductive, with mutual inductance L_{01}. The impedance signal measured by the readout equipment is

$$Z_{\text{tot}}(\omega) = Z_0(\omega) + \omega^2 L_{01}^2 / Z_1(\omega). \tag{1}$$

A plot of $|Z_{\text{tot}}|$ is shown in Fig. 4a. The lower the ratio R_1/L_1, the sharper the peak. Note that the response curve can become more complicated if non-ideal geometrical effects are taken into account.

Experimental hardware. We present some details of the test circuits we developed (see Fig. 3a). They were made using thin film deposition methods on 6 inch glass wafers. The circuits contain two conductive layers separated by a thin layer of randomized dielectric. The lower layer is 1μm thick Al, containing only a capacitor plate. The upper layer is 10μm thick Cu, containing the opposite capacitor plate and a coil. The area of the coil is slightly less than 1mm^2. In order to have a good antenna functionality, the coil sits at the perimeter of the available area. Additional coil windings closer to the center would not improve the signal strength, while adding to the resistance. The resonance frequencies vary between 200 MHz and 1.6 GHz.

The complex impedance was measured using a spectrum analyzer. The readout coil is made of a few windings of copper wire, with area comparable to the PUF coil area. The wire is shielded by a conical sheet of Cu (see Fig. 3b). The distance between the circuit and the readout coil is between 0.5 mm and 1 mm.

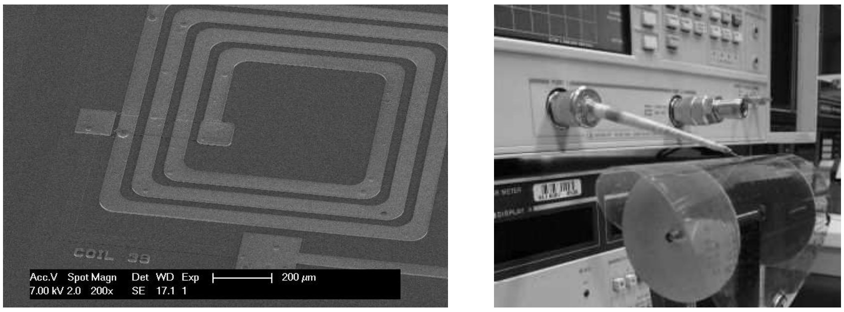


Fig. 3. (a) Upper metal layer of an LC-PUF. (b) Spectrum analyzer and readout coil; circuits on a flexible substrate for bending tests.

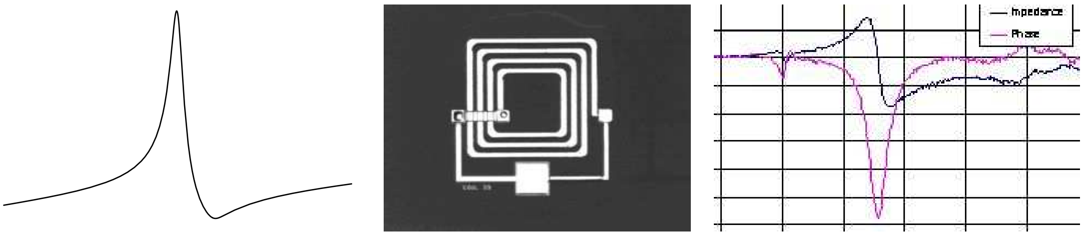


Fig. 4. (a) Theoretical response curve $|Z_{\text{tot}}|$ for ideal components. (b) Top view of an LC-PUF. (c) Response curve of that LC-PUF.

Experimental results. A typical response curve is shown in Fig. 4c. We studied the amount of noise under repeated repositioning in the (x,y) plane. At constant temperature, the reproducibility of the upward peaks is better than 0.15 MHz (standard deviation) for the low frequency peaks, and better than 0.3 MHz for the high frequency peaks. When vertical repositioning errors are considered as well, these numbers change to 0.3 MHz and 1 MHz, respectively. It is important to note that the production spread in thin film deposition is often larger than these values. Thus, it is hard to clone these structures in practice. We also studied the effect of temperature between 25°C and 75°C. We found a monotonous decrease of the resonance frequencies of less that 1% over this whole temperature range. Hence temperature effects are easy to compensate. As expected, Fig. 5 shows how the response of two different LC-PUFs is different. We performed tests in 500 chips and based on these tests, we conclude that the response curve of a single randomized LC-resonator of simple design is equivalent to an identifier string with a length of approximately 9 bits (pessimistic estimate of repositioning accuracy) to 11 bits (optimistic). More bits can be obtained by constructing more complicated circuits.

4.4 Intrinsic PUFs and SRAM memories

The disadvantage of most of these approaches is the fact that custom built circuits are used or that a modification of the IC manufacturing process is required. In Guajardo et al. (2007b), the authors introduce *Intrinsic* PUFs which are defined as PUF generating circuits already present in the device requiring little or no modification to satisfy the security goals. Intrinsic PUFs were introduced in Guajardo et al. (2007b), where it is shown that the start-up values of SRAM memories present in FPGAs work well as PUFs. The behavior of SRAM memories as PUFs, however, is not expected to be limited to FPGAs. In Holcomb et al. (2007), a similar idea is presented but this time the device under study was an ultra-low power chip used in sensor node

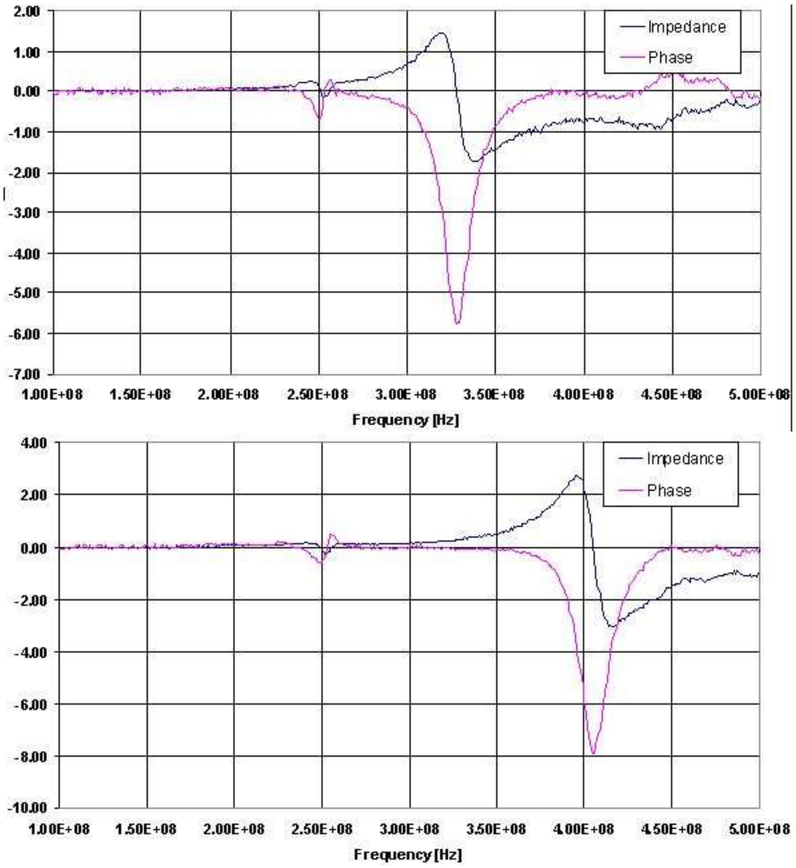


Fig. 5. Frequency response peaks corresponding to two different LC-PUFs

applications. In the following, we summarize the ideas of Guajardo et al. (2007b) and describe why start-up values of SRAM memories are essentially a PUF.

We begin this section by describing the structure of a typical six transistor CMOS SRAM cell Bellaouar and Elmasry (1995) as shown in Fig. 6. Such a cell consists of two cross-coupled inverters (load transistors PL, PR, NL and NR) and two access transistors (AXL and AXR) connecting to the data bit-lines (BLC and BL) based on the word-line signal (WL). Each inverter consists, in turn, of a p-junction transistor (PL, PR) and an n-junction transistor (NL, NR). A key characteristic of an SRAM cell is the static-noise margin (SNM), defined as the minimum DC noise voltage to flip the cell state. In fact, much research is aimed at optimizing the SNM while at the same time reducing the size of the cell (which tend to be opposing aims). Optimizing (increasing) the SNM results in a more stable cell, thus requiring a higher voltage to flip the state of the cell. Notice that the SNM, in turn, has been shown to be directly influenced by the threshold voltage of the cell's transistors (Seevinck et al., 1987). Other variations affecting the threshold voltage V_t of the transistors of an SRAM cell have been studied in Bhavnagarwala et al. (2001) (see also Cheng et al. 2004). Such variations result in different SRAM cells in a SRAM memory array having slightly different threshold voltages and as a result different SNMs. Since such variations are well known to occur, memory designers construct SRAM cells with proper width/length ratios between the different transistors (Seevinck et al., 1987). This guarantees that known variations outside their control do not affect the reading and writing process of the cell under normal operation. However, during power-up, the SRAM cell's cross-coupled inverters are "floating". Therefore, the previously discussed SNM differences in the transistors will cause the

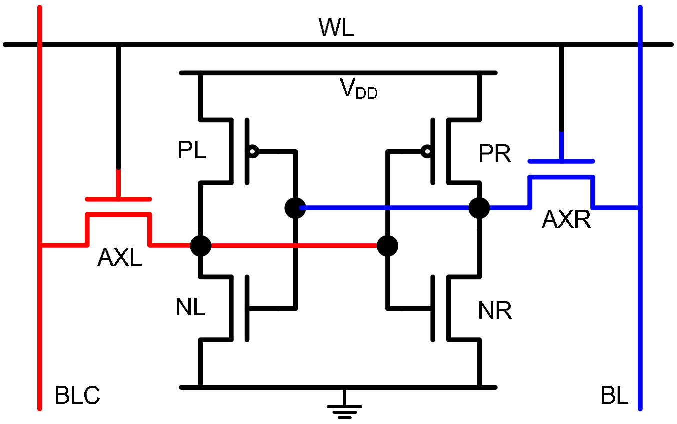


Fig. 6. Six transistor SRAM cell

stored value to tend toward a 0 or a 1, depending on the cell's specific characteristics. This effect is increased by the amplifying effect of each inverter acting on the output of the other inverter. As a result (and as shown in Guajardo et al. 2007b; Holcomb et al. 2007), the same SRAM cell will tend to start in the same state upon power-up whereas different SRAM cells will behave randomly and independently from each other. Thus SRAM memory arrays form an Intrinsic PUF. As noticed in Guajardo et al. (2007b), one can consider as a challenge a range of memory locations within a SRAM memory block. An additional advantage of SRAM-based PUFs is that their responses are immediately in binary form (Guajardo et al., 2007b). This is in contrast to previously reported PUFs in which a quantization step is needed to turn an analog measurement into a binary response. Hence, the complexity of the measurement circuit is reduced. We refer to (Guajardo et al., 2007a,b) for a detailed discussion of the properties of SRAM-based IPUFs. Notice that SRAM IPUFs have excellent identification properties, exhibiting high entropy and acceptable measurement noise across a wide range of temperatures (Guajardo et al., 2007b). The authors estimated that to be able to extract a 128-bit key, about 5000 SRAM memory cells are required, because of noise and randomness requirements.

5 Secure key deployment for sensor nodes with PUFs

As previously mentioned and noted recently in Kuo et al. (2007), many protocols assume some secure way of transferring an initial key to a wireless device without actually specifying how to accomplish such a task. The starting point for the protocols that we propose in this paper is the Message-In-a-Bottle (MIB) protocol of Kuo et al. (2007) as this is, to our knowledge, the only work that has actually addressed the problem in a thorough manner (see also Sect. 5.4). As we will see a basic building block in the protocol is a Faraday cage which provides privacy from eavesdroppers during the initial key set-up. We propose to depart from this approach and to use PUFs and the corresponding helper data algorithm as a secure manner to initialize the sensor node. These protocols will be explained in detail in Sect. 5.3. Before continuing, we summarize the attacker model and services offered by the MIB protocol. This will allow us to make a fair comparison later in Sect. 5.5.

5.1 Assumptions and strong attacker model

The problem at hand is best explained by the example provided in Kuo et al. (2007): a customer receives a shipment of new sensor nodes and using a wireless communication channel (no other channel is assumed because of cost considerations) he wants to set a shared secret between the uninitialized nodes and a wireless base station. Kuo et al. (2007) provide a list of properties that a solution for this problem should provide:

1. **Key secrecy:** an attacker has negligible chance of compromising the shared secret between nodes and base station.
2. **Key authenticity:** an uninitialized node receives the key that the base station originally sent and not a key coming from an adversary.
3. **Forward secrecy:** compromising one node does not compromise the keys on previously deployed nodes. More importantly, the attacker that compromises a node only gets knowledge to the current key and knows nothing of previously generated keys.
4. **Demonstrative identification:** users physically handle devices in such a way that they are certain of which devices are communicating.
5. **Robust to user error:** the system should be designed around users and for normal users (not expert cryptographers or security engineers). In addition, a user error should not result in key compromise.
6. **Cost effective:** the proposed solution should not add to the costs of the sensor node and/or of the network.
7. **No public-key cryptography:** in general public-key (PK) cryptography implementations are more expensive in terms of program space, slower speed and, if implemented at the hardware level, silicon area. In addition, PK crypto can make nodes susceptible to energy draining Denial-Of-Service (DOS) attacks. Notice that although this is the case at the present moment, we expect that some sort of PK crypto will be possible in wireless sensor networks eventually. For example, as early as 2001, there has been implementations of PKC on ultra-low power micro-controllers used today for wireless sensor applications, see e.g. Guajardo et al. (2001).

As in Kuo et al. (2007), we also assume that installation personnel can be trusted and that they can follow simple instructions (as in a cooking recipe). Similarly, we assume that once an initial key has been set up, the nodes will use secure communication protocols. The work of Kuo et al. (2007) also assumes that other devices (keying device and keying beacon) are present to facilitate the key deployment. We will show that the required number of devices used during key deployment is reduced in one of our protocols compared to MIB and thus, that our solution is more cost efficient. In our solution, we also assume that there is a PUF present in the sensor node with its corresponding security properties as described in Sect. 3.2. Notice that the presence of such a PUF does not necessarily increase the cost of the node as Intrinsic-PUFs are inherently present in silicon devices as shown in Guajardo et al. (2007b) and independently on an ultra-low power micro-controller in Holcomb et al. (2007). In any event, we expect the costs to be minimal.

Finally, we assume a very powerful adversary, whose aim is to compromise the keys to be shared by the nodes and the base station. The attacker can overhear, intercept, and inject any messages into the communication channel. In this model, we also assume (as in Kuo et al. 2007) that the attacker is omni-present, i.e. the attacker is present before, during, and after key deployment.

5.2 The Message-In-a-Bottle secure key deployment protocol

We describe in some detail the Message-In-a-Bottle key deployment protocol (Kuo et al., 2007), as our new protocol can be seen as a modification of some sub-protocols in MIB. In MIB five different parties participate:

(i) the base station (S), which controls the entire network and has the capabilities of a regular PC. The base station delegates key deployment to less powerful devices (the keying device and the keying beacon) by transferring the keying material via a secure channel such as a secondary physical USB interface.

(ii) the sensor node (M), which is to share a secret with the base station. Upon being powered on or reset the node is in an uninitialized state, once the node receives the correct key, the node changes to an initialized state. Finally, if the key deployment fails, the node is in a rejected state without sharing a valid key with the base station.

(iii) the keying device (D), which is placed inside a Faraday cage together with the node and sends the initial key information to the node when the Faraday cage is closed. Then, the node uses the keying information received from the keying device to derive the key.

(iv) the keying beacon (B), which is used to signal that the Faraday cage is closed and to jam the communication channel (outside the Faraday cage), thus preventing an eavesdropper from obtaining information leaked by the Faraday cage. In addition, the keying beacon provides the user with status information about the deployment and its outcome.

(v) the user, who wants to perform the key deployment.

The protocol assumes weak time synchronization between D and B. This is achieved via counters and authenticated messages between the devices. The authentication protocol uses a keyed Message-Authentication-Code (MAC). The key K_D used to derive the deployment key for node M is generated by the base station and securely transmitted to D. Then, for every new node M, the following steps are followed:

1. Once D and B have exchanged authenticated synchronization messages, the user turns on the node M and places D and M inside a Faraday cage. The user then closes the Faraday cage, the keying beacon is outside the Faraday cage and unable to communicate with D.
2. Inside the Faraday cage, D generates the node's M key as a pseudo-random function of the node's ID M, keyed with the current value of K_D, i.e. $K_M = \mathrm{PRF}_{K_D}(M)$. The keying device updates $K_D \leftarrow \mathsf{Hash}(K_D)$, increases a counter c by one (this is used to keep track of how many times K_D has been updated), computes $h \leftarrow \mathsf{Hash}(K_M)$ and sends h to M. The value h works as a commitment to K_M, which M can use later to verify validity of K_M. Notice that updating K_D via hashing ensures that a new key is used for every new node M and provides forward secrecy
3. D generates s random nonces $r_1, r_2, ..., r_s$, computes the activation key $k = K_M \oplus r_1 \oplus \cdots \oplus r_s$ and sends the r_i's to M over s rounds of communication along with the counter c. Thus, an attacker must overhear all s messages in order to compromise the key K_M.

In addition to the above mentioned tasks, the keying device D monitors the amount of noise in the background. If the Faraday cage is left open or it is not attenuating signals as expected, D will detect the presence of the keying beacon and abort the deployment. At the same time, the keying beacon jams all communications in the frequency of the deployment during the few seconds that the key deployment takes place. After such a short period of time, B signals the user to open the Faraday cage, and once the keying beacon and keying device verify that the protocol was performed as expected, the keying device sends the value of k (the validation key) to M, which then computes K_M and verifies that h (received in Step 2 of the protocol) corresponds to $\mathsf{Hash}(K_M)$. The end of the protocol verifies the correctness of the deployed keys by computing a MAC on the value k keyed with K_M. We refer to Kuo et al. (2007) for the details as they are not relevant to the discussion here.

The MIB protocol ensures that any user errors (like an open Faraday cage or too early opening) only leads to an erroneous key rather than to key leakage. This method requires no additional hardware per sensor node. However, it does require additional specialized hardware: *Keying Device*, *Keying Beacon* and a Faraday cage. Both, ease of use and robustness are achieved in the MIB protocol thanks to the ability of users to physically manipulate devices in such a way that they are certain which devices are communicating.

As noticed in Kuo et al. (2007) one could solve the key deployment problem by using manufacturer installed keys. However, Kuo et al. argue that this is not a good idea because of three main reasons:

1. There is no assurance that an attacker did not tamper with the hardware anywhere along the distribution chain: from manufacturer to end user.
2. Customers would have to trust the manufacturer to manage keys in a proper and secure manner.
3. Manufacturers do (might) not want to assume liability for key management.

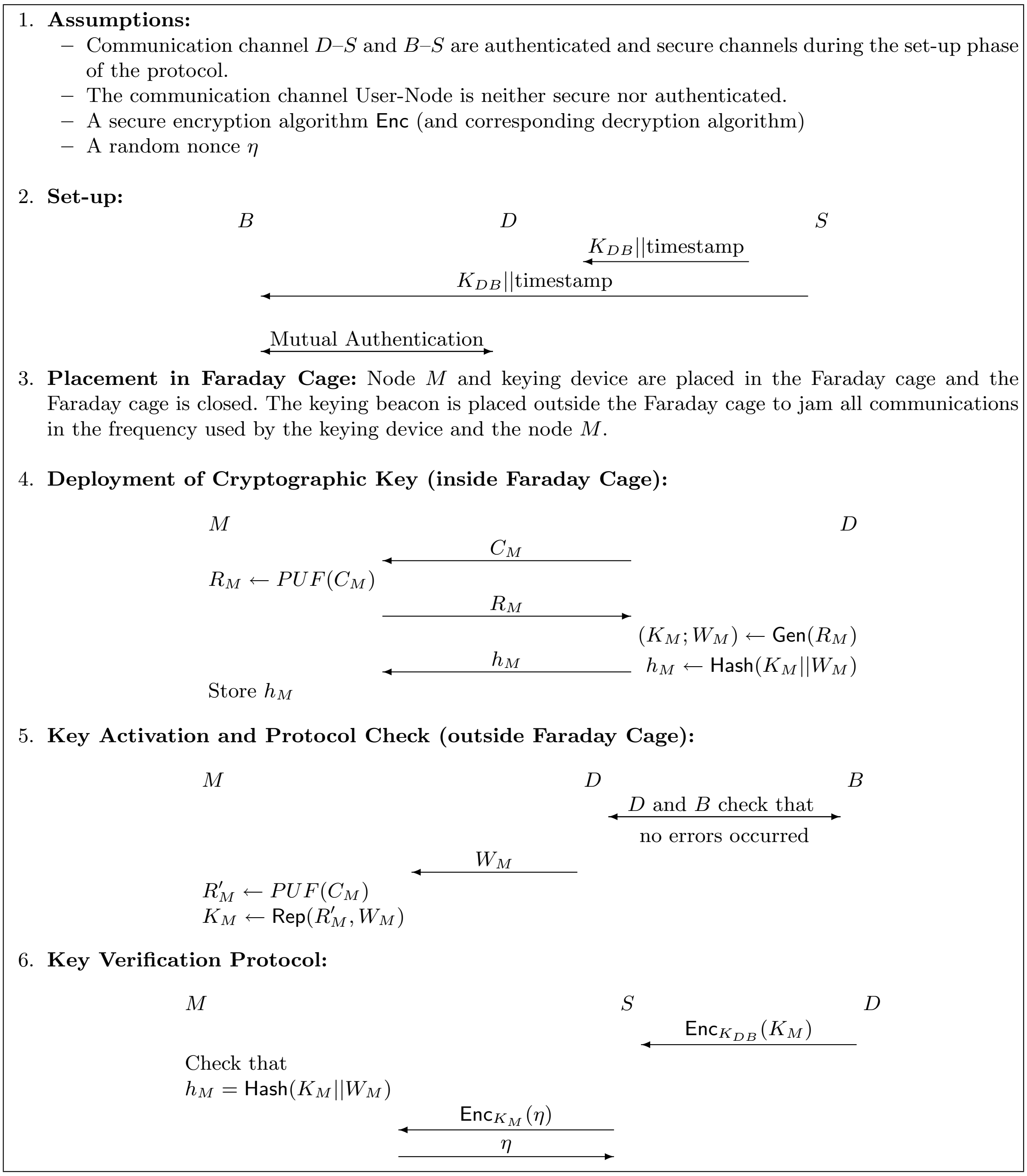
In the next section, we show that the problem of tampering and tamper evidence can be easily solved with PUFs. In addition, we show protocols that require minimal trust on the manufacturer or more generally a trusted third party (TTP). The final argument of not accepting liability is, in our view, more of a subjective one. Whether the manufacturer or the TTP want to assume liability will depend on the application, business model, etc. Certain TTPs might be interested in doing it while others might not. Thus, we do not consider such argument in the remainder of the paper.

5.3 PUFs, fuzzy extractors and their use for key deployment

In this section, we present two protocols allowing secure key deployment to uninitialized sensor nodes. We make a distinction between two situations. The first protocol is similar in nature to MIB but we modify the key activation part at the end of MIB by using helper data as the activation key. We notice that the use of a secure area somewhere in the overall protocol seems to be a must. In other words, unless there is at some point in time an area in which the attacker can not eavesdrop, preserving key confidentiality seems unattainable. Thus, in the first protocol, we also make use of a Faraday cage. Notice that the first protocol provides the same guarantees as the MIB protocol, with reduced communication complexity and the added advantages of using a PUF. In other words, a PUF provides unclonability and (depending on the PUF) tamper evidence.

In the second protocol, we assume the existence of a Trusted Third Party (TTP). This can be the manufacturer or a different entity charged with the authority of distributing and managing keys. We also show ways in which trust on the TTP in a “real-world” scenario (i.e. a world where an attacker can not be present everywhere) can be reduced to preserve the confidentiality of communications between sensor nodes and base station. This implies that in the second protocol we assume a weaker attacker model. In particular, the TTP has knowledge of the key and the hardware manufacturer can gain knowledge of the key if it is present both during the enrollment phase and during key deployment eavesdropping at the end-user premises. In such a weaker security model, we show that the use of PUFs allow for a significant reduction in the protocol complexity and a significant reduction in the hardware required for deployment. In addition, as in the first protocol, using PUFs provides unclonability and tamper evidence as well.

Secure deployment without trusted third parties in the strong attacker model. To achieve secure key deployment without a trusted third party, we require the same hardware that the MIB protocol requires: the base station (S), the keying device (D), the keying beacon (B), and the sensor node (M). In this case the hardware manufacturer has no involvement in the protocol. The protocol is shown in Figure 7. We present the overall protocol for completeness.



1. **Assumptions:**
   - Communication channel D–S and B–S are authenticated and secure channels during the set-up phase of the protocol.
   - The communication channel User-Node is neither secure nor authenticated.
   - A secure encryption algorithm Enc (and corresponding decryption algorithm)
   - A random nonce η
2. **Set-up:**

3. **Placement in Faraday Cage:** Node M and keying device are placed in the Faraday cage and the Faraday cage is closed. The keying beacon is placed outside the Faraday cage to jam all communications in the frequency used by the keying device and the node M.
4. **Deployment of Cryptographic Key (inside Faraday Cage):**

M
D
C_M
$R_M \leftarrow PUF(C_M)$
R_M
$(K_M; W_M) \leftarrow \mathsf{Gen}(R_M)$
h_M
$h_M \leftarrow \mathsf{Hash}(K_M||W_M)$
Store h_M

5. **Key Activation and Protocol Check (outside Faraday Cage):**

M
D
B
D and B check that
no errors occurred
W_M
$R'_M \leftarrow PUF(C_M)$
$K_M \leftarrow \mathsf{Rep}(R'_M, W_M)$

6. **Key Verification Protocol:**

M
S
D
$\mathsf{Enc}_{K_{DB}}(K_M)$
Check that
$h_M = \mathsf{Hash}(K_M||W_M)$
$\mathsf{Enc}_{K_M}(\eta)$
η

Fig. 7. Key deployment protocol without TTP

The set-up phase of the protocol is essentially the same as in MIB. The mutual authentication between the keying device and the keying beacon guarantees that both devices are not subject to a man-in-the-middle-attack[6] whereas the timestamps are used for weak synchronization between the devices. This way during the key activation phase of the protocol, both D and B can check that indeed the keying beacon was jamming the communication channel during key deployment and that the Faraday cage was closed.

The key deployment protocol is essentially performing the enrollment protocol as in Sect. 3.3. In addition, it stores a hash of the key and the helper data in M, which during the key validation and verification phase can be used by the node to check the validity of the activated key. The activation protocol is the similar as in the TTP-based protocol, i.e., the helper data W_M is sent over to the node (in the clear) and the node then constructs the key K_M. Notice that no information about the key is disclosed by sending W_M in the clear thanks to the fuzzy extractor constructions described in Sect. 3.3. During the verification phase the node M checks the validity of the hash value received during the key-deployment phase and proves to the base station that it is in possession of a valid key following the key verification protocol outlined in Fig. 7. An advantage in the current protocol is that the node does not require the presence of a random number generator to check the validity of its key. This translates into more space for performing other tasks, storing additional application code, or reduced hardware costs. Finally, notice that in the MIB protocol security is somewhat enhanced by splitting the key into shares and transmitting the key shares over an extended period of time. This forces an adversary to be able to obtain all shares to successfully compromise the key. Similar techniques can be applied in our protocols if deemed necessary. In particular, instead of sending R_M in a single message, M could do this by computing $R'_M = R_M \oplus r_1 \oplus \cdots \oplus r_s$ and send the values $R'_M, r_1, \ldots, r_s$ one after another. This, however, would require the presence of a random number generator in the sensor M. The security of such scheme can be further enhanced by making the transfer of shares in a time delayed manner (see Bird et al. 2007 for a description of a similar scheme in the RFID context). Such scheme has the advantage of not requiring a random number generator in M.

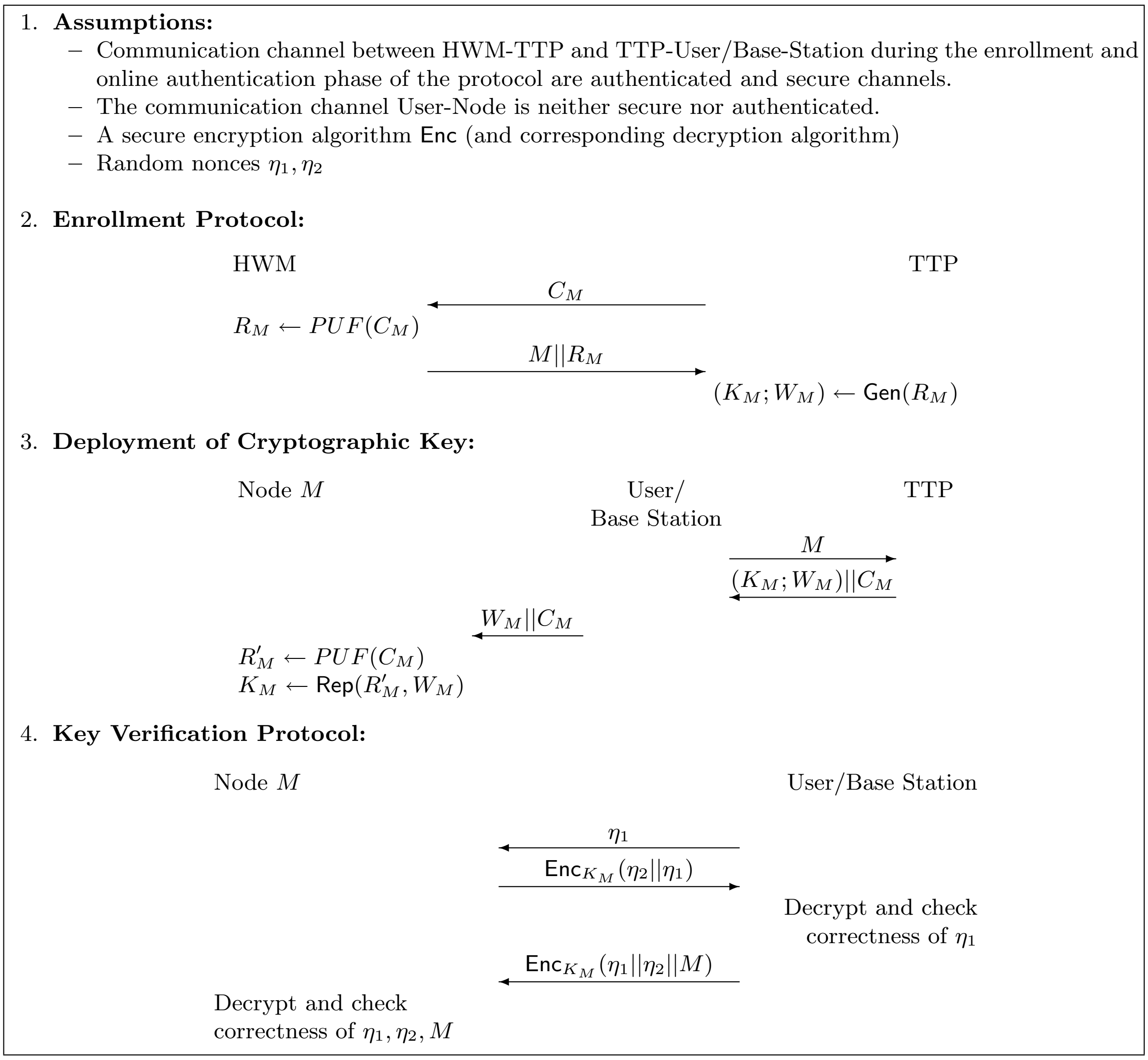
Secure deployment with trusted third parties in a weaker attacker model. In this protocol we assume the existence of a TTP. The protocol begins with a trusted third party performing an enrollment protocol by running $(K_M; W_M) \leftarrow \mathsf{Gen}(R_M)$ on the PUF response R_M as explained in Sect. 3.3. Observe that the TTP can be the hardware manufacturer (HWM) of the sensor nodes itself or an independent third party. The advantage of having an independent TTP is that the key is only known to the TTP and the end-user and not to the manufacturer. This is true since both R_M and W_M are necessary to reconstruct the correct key K_M and the manufacturer only knows R_M. Thus, we assume implicitly that the hardware manufacturer is not omnipresent. In particular, if desired, the HWM could eavesdrop the deployment of the helper data W_M during key verification, thus gaining knowledge of the key K_M. This implies that we are working in a weaker attacker model (or alternatively, we trust the HWM). Notice, however, that this weaker attacker model provides us with significant reductions in both protocol complexity and hardware resources (i.e. cost) when compared to the original MIB protocol.

The values $(K_M; W_M)$ corresponding to node M are then sent to the user via a secure and authenticated channel. When the user receives the node and associated $(K_M; W_M)$ values, these are installed in the base station as corresponding to node M. Then, the following steps are performed:

[6] As in Bellare and Rogaway (1993), we do *not* consider it to be an attack if the adversary only relays messages between the intended parties as this can not be prevented. In this case, (as noted in Bellare and Rogaway 1993) the adversary is simply acting as a wire. Thus, a man-in-the-middle attack requires modification of the messages as well.

1. The base station sends in the clear the value W_M to the node M.
2. Node M measures the PUF and obtains a response R'_M. Then, node M performs the information reconciliation and privacy amplification procedures described in Sect. 3.3, thus reconstructing the key $K_M \leftarrow \mathsf{Rep}(R'_M, W_M)$. Notice that the helper data has the same function as the activation key k in MIB.
3. The base station and the node then engage in a mutual authentication protocol such as the one suggested in Kuo et al. (2007) to verify the correctness of the installed key. Any other challenge-response protocol for mutual authentication can be used as well, e.g., see Menezes et al. (1997, Chapter 10).

The overall protocol is shown in Figure 8. We show an instantiation of the key verification part of the protocol based on standard mutual authentication techniques based on symmetric encryption. Notice, however, that similar protocols exist, which are based on Message-Authentication-Codes (MACs) or keyed hash functions (Menezes et al., 1997, Chapter 10).



1. **Assumptions:**
   - Communication channel between HWM-TTP and TTP-User/Base-Station during the enrollment and online authentication phase of the protocol are authenticated and secure channels.
   - The communication channel User-Node is neither secure nor authenticated.
   - A secure encryption algorithm Enc (and corresponding decryption algorithm)
   - Random nonces η_1, η_2

2. **Enrollment Protocol:**

HWM
TTP
$\xleftarrow{\quad C_M \quad}$
$R_M \leftarrow PUF(C_M)$
$\xrightarrow{\quad M||R_M \quad}$
$(K_M; W_M) \leftarrow \mathsf{Gen}(R_M)$

3. **Deployment of Cryptographic Key:**

Node M
User/
Base Station
TTP
$\xrightarrow{\quad M \quad}$
$\xleftarrow{\quad (K_M; W_M)||C_M \quad}$
$\xleftarrow{\quad W_M||C_M \quad}$
$R'_M \leftarrow PUF(C_M)$
$K_M \leftarrow \mathsf{Rep}(R'_M, W_M)$

4. **Key Verification Protocol:**

Node M
User/Base Station
$\xleftarrow{\quad \eta_1 \quad}$
$\xrightarrow{\quad \mathsf{Enc}_{K_M}(\eta_2||\eta_1) \quad}$
Decrypt and check correctness of η_1
$\xleftarrow{\quad \mathsf{Enc}_{K_M}(\eta_1||\eta_2||M) \quad}$
Decrypt and check correctness of η_1, η_2, M

Fig. 8. Key deployment protocol with TTP

It is clear that one disadvantage of the protocol is that the TTP knows the deployed key for the specific node M. However, this might be outweighed by the fact that our protocols do not require any additional hardware (i.e. no key beacon or keying device are required) and the protocol can be performed without having to introduce the nodes into a Faraday cage. Notice that the Faraday cage is still present at the manufacturer's side. However, in our protocol, it is the manufacturer or the TTP who have to invest in such secure facility, which we consider plausible. This can be achieved via a combination of infrastructure (Faraday cage present somewhere in the manufacturing process) and physical access control mechanisms (police, guards, secure facilities, etc.). The question of tampering with the device during transit (between the manufacturer and the end-user) is also of lesser concern since PUFs guarantee tamper evidence, tamper resistance (e.g. coating PUFs) and unclonability (all PUFs). In addition, if the attacker was to tamper with the PUF, the mutual authentication step at the end of the protocol would fail since the Rep procedure would generate a different key from the one generated (and sent to the end-user) by the TTP. Regarding forward secrecy, there is no universal key stored in the base station from which node keys are derived. Every node has a different key and compromising any node's key does not give any information about a different node's key.

The verification step requires that the node M be able to generate a random nonce η_2. Low-power random number generators have been proposed in Perrig et al. (2002); Castelluccia and Francillon (2007). Both approaches make use of pseudo-random number generators based on a keyed MAC algorithm and an incrementing counter. That poses the question of what key (call it K_{RNG}) to use. Since we are only using the key to generate a random number, one could simply use the deployed K_M for that purpose. If for some reason, the key had been tampered with, then the verification protocol will fail on the user/base-station side and the user can take appropriate measures. An alternative is to use $K_{RNG} \leftarrow \mathsf{Hash}(K_M||i)$, where i is a random bit and Hash is a collision resistant hash function. The random bit could originate from the PUF response which is only available within the sensor node. This would prevent the attacker from choosing a key that creates a known nonce η_2, since the attacker is then not able to predict the value of K_{RNG}. Another possibility is to use a random number generator based on PUFs as described in O'Donnel et al. (2004). Finally, we notice that it is also possible to achieve the verification without generating a random nonce in the node. This can be achieved by adding a second round of communication in which the TTP sends the value $h_M \leftarrow \mathsf{Hash}(K_M||W_M)$ to the HWM and the HWM stores it in the node M. No knowledge of the key K_M is disclosed thanks to the properties of the hash function. An attacker could tamper with the value h_M and with W_M during the verification phase of the protocol. However, the verification step will nevertheless fail since it depends on knowledge of the key K_M and of the hash value h_M, neither of which the attacker can compute thanks to the properties of hash functions and of secure fuzzy extractors (i.e. you obtain negligible information about K_M from W_M).

5.4 Related work on secure key deployment for sensor nodes

Though there has been a lot of work on different key deployment schemes for sensor networks (such as ZigBee Specification 2005, SPINS (Perrig et al., 2002), LEAP (Zhu et al., 2003, 2006), Transitory Master Key (Deng et al., 2005), and random key pre-distributions (Chan et al., 2003; Du et al., 2003; Eschenauer and Gligor, 2002; Liu et al., 2005; Ramkumar and Memon, 2005)), most of them assume the initial secret key to be on the sensor node based on an unspecified security mechanism. However, there are also other key establishment procedures which address the initial key exchange like the Message-In-a-Bottle (Kuo et al., 2007), Resurrecting Duckling (Stajano and Anderson, 1999; Stajano, 2000), Talking to Strangers (Balfanz et al., 2002), Seeing-is-Believing (McCune et al., 2005), On-off Keying (Cagalj et al., 2006), Key Infection (Anderson et al., 2004), and Shake Them Up (Castelluccia and Mutaf, 2005). These protocols differ on various security considerations, ease of use and associated costs.

Using an out-of-band channel physical contact is the method that is used in the Resurrecting Duckling to securely share a secret key (Stajano and Anderson, 1999; Stajano, 2000). This method can securely and authentically share a secret-key between devices if the direct contact channel is assumed to be secure. It also gives demonstrative identification and is robust to user errors. However, the main disadvantage is the need for extra hardware per-node to enable the information exchange through a physical contact. Another out-of-band channel based method is Talking-to-Strangers which uses a location-limited channel like infrared or audio to setup a public key (Balfanz et al., 2002). This method is similarly not cost effective both due to the need for extra specialized hardware per sensor node for the communication and the use of public-key cryptography.

Seeing-is-Believing methods use a public-key encoded as a 2D-barcode to set up the key (McCune et al., 2005). Unlike the Talking-to-Strangers protocol, Seeing-is-Believing requires only a single specialized set-up hardware equipped with a camera or barcode reader. However, it does require costly public-key cryptography to be performed on the sensor nodes. The Shake-Them-Up scheme sets up keys among nodes by holding a node in each hand and shaking them. The nodes exchange identical packets and thus, the attacker is not able to distinguish between messages originating from either device and transmitted on the same wireless channel (Castelluccia and Mutaf, 2005). To avoid the attacker spatially distinguishing the sources based on the power, the devices are shaken together during this communication. This approach, however, is not fully secure due to radio fingerprinting (Rasmussen and Capkun, 2007). Though it provides physical identification of the devices, the key could be compromised if the user does not shake sufficiently. Smart-Its Friends (Holmquist et al., 2001) and Are-You-with-Me (Lester et al., 2004) are related schemes but requiring additional accelerometer on the nodes to measure movement.

The On-off Keying technique uses the presence or absence of the RF signal to encode a 1 or a 0 respectively (Cagalj et al., 2006). Assuming the attacker could only modify a 0 (RF absence) to 1 with an RF signal, then the message can still be authenticated by encoding it appropriately. Authenticity cannot be completely guaranteed as the authors of the scheme do not specify how the devices know what the authentic levels of 1 and 0 are. The scheme also requires the use of public-key cryptography and lacks a physically demonstrative identification of the devices with which keys are shared. Key Infection is just a simple and cost effective scheme assuming that the attacker is not present at the moment the keys are shared (Anderson et al., 2004). Hence, the keys are sent in the clear which breaks both the security and authenticity because the key exchange could be performed also by an adversary. Clearly, such a scheme contradicts the security model in which it is assumed that the attacker is present before, during and after the key set-up procedure.

5.5 Comparison

Kuo et al. (2007) provided an extensive comparison of their protocol with previous ones in Table 2 of their work. Thus, we augment their table with our two new protocols. We also add to the table the category tamper evidence and unclonability. The resulting table is shown here as Table 1.

One can argue that our solution with a TTP does not provide key secrecy in the same sense that MIB or our solution without TTP. However, it would also not be adequate to say that it offers no key secrecy since the only eavesdroppers that can compromise the key are the TTP and the hardware manufacturer which, depending on the application, are trusted. In addition, for the key to be compromised by the HWM, the HWM has to be active, i.e., it should be actively

	This paper (with TTP)	This paper (without TTP)	Message-In-a-Bottle (Kuo et al., 2007)	Resurrecting Duckling (Stajano and Anderson, 1999)	Talking to Strangers (Balfanz et al., 2002)	Seeing-is-Believing (McCune et al., 2005)	On-off Keying (Cagalj et al., 2006)	Key Infection (Anderson et al., 2004)	Shake Them Up (Castelluccia and Mutaf, 2005)
Security									
Key secrecy	Y*	Y	Y	Y	–	–	–	N	N
Key authenticity	Y	Y	Y	Y	Y	Y	N	N	Y
Key unclonability and tamper evidence	Y	Y	N	N	N	N	N	N	N
Usability									
Demonstrative identification	Y	Y	Y	Y	Y	Y	N	N	Y
Robust to user error	Y	Y	Y	Y	Y	Y	Y	Y	N
Costs									
No per-node extra hardware	Y	Y	Y	N	N	Y	Y	Y	Y
No specialized set-up hardware	Y	N	N	Y	Y	N	Y	Y	Y
No public-key cryptography	Y	Y	Y	Y	N	N	N	Y	N

Table 1. Comparison of different key deployment techniques based on Kuo et al. (2007). A '–' signifies hat this property is not applicable.

trying to eavesdrop the communications of the user and be present during key deployment. This is in sharp contrast with other protocols (Anderson et al., 2004; Castelluccia and Mutaf, 2005) in which *any* eavesdropper can compromise the secrecy of the key.

In addition, PUFs provide another type of security guarantee implied by their unclonability and tamper evidence. Such property is only available to PUF-based solutions. PUFs also provide simplifications in the protocols. This is particularly true if we look at the number of rounds of communication in our newly proposed protocols and compare this number to those of the MIB protocol. In the case of the TTP-based protocol, PUFs also allow to get away without any specialized set-up hardware, which will certainly reduce costs.

It is also important to point out the advantages that a PUF-based solution has over a solution based on burning the key in the node's ROM memory. Such a ROM-based solution allows the HWM to know the key without any effort and provides no guarantees as to whether the key has been tampered with by the time the end-user gets the sensor node.

6 Other PUF applications

6.1 IP protection on reconfigurable hardware

The main example of reconfigurable hardware that we consider in this paper are S-RAM (Static RAM) Field Programmable Gate Arrays (FPGAs). Essentially they can be thought of as configurable hardware that can be programmed to carry out specific functionality. They are very popular for several reasons: i) the upfront investment cost is very low compared to that of ASICs and ii) they are very flexible since they can be reconfigured in the field. In order to program a FPGA, a bitstream that embeds its functionality has to be developed. The bitstream is stored in external memory (*e.g.* PROM). At power-up, the bitstream is then transmitted to the FPGA. Once loaded the FPGA is configured and ready to carry out its functionality. We stress that most of the value is in the bitstream. Indeed when the bitstream is copied and stored in the external memory of another FPGA, another chip with the same functionality is obtained. Since, the bitstream is often loaded without any protection from the external memory to the FPGA it is relatively easy for an attacker to capture the bitstream and make a copy without further research and development costs. This attack which is easy to carry out, is nowadays called the *cloning* attack.

Clearly encryption of the bitstream with a key that is specific to a particular FPGA would solve the problem. This observation is due to Kean (2002), who also proposes an associated protocol to support IP protection. The protocol is based on bitstream encryption using a key stored in non-volatile memory on the FPGA. One general problem with this solution is that there is no non-volatile memory on the vast majority of SRAM FPGAs to store a long-term key. In order to solve this problem two main solutions have been proposed: (i) some non-volatile memory such as flash is added to the FPGA and (ii) the FPGA stores a long-term key in a few hundred bits of dedicated RAM backed-up by an externally connected battery. It is clear that the previously mentioned solutions come with an additional cost. The second solution has the additional disadvantage that the battery has only a limited life time and that batteries can get damaged which shortens further their life-time. In addition, certain problems can not be easily solved via bitstream encryption alone.

For example, Simpson and Schaumont (2006) have identified two potential problems if the aim of the solution is to secure third party intellectual property and software modules. These are: (i) Intellectual Property (IP) authentication by system (SYS) developers as well as authentication of the hardware platform (where the software IP is running) by the IP providers (IPP) and (ii) protection of the software that is running on the processors configured on the FPGA. Several other works (Kahng et al., 1998; Kean, 2002; Guajardo et al., 2007a) have identified other security services of interest in the IP value chain which can be envisioned between the different parties involved in the chain, from hardware manufacturer (HWM) to End User. These services are summarized in Table 2.

The authors in (Simpson and Schaumont, 2006) are the first to suggest the use of a PUF to provide such services. In Guajardo et al. (2007b) the authors simplify the protocols of Simpson and Schaumont (2006). The basic idea in both works is to bind the IP to be protected (i.e. the FPGA configuration file) to the FPGA via a PUF. In particular, the configuration file is stored in insecure non-volatile memory in encrypted format. Upon power-up, the FPGA reads the encrypted configuration file, challenges its PUF and reconstructs the key used to encrypt the configuration file with a helper data algorithm (as explained in Sect. 3.3), decrypts the configuration file, and configures the FPGA. The authenticity of the data is checked via a keyed Message Authentication Code (MAC) with a PUF derived key. In Guajardo et al. (2007a), this work is generalized to the public-key setting and it is shown that if we assume the existence of a public-key cryptographic processor on the FPGA, the secret-key does not need to leave the FPGA (even during enrollment) and thus, secrecy is provided even from an honest-but-curious TTP.

Table 2. Security Services in the IP Protection Chain

Security Service	Description
IP authenticates Hardware	IP can only be executed on one specific hardware device, hence it can not be cloned.
Hardware authenticates IP	The hardware platform (FPGA) detects tampering with the IP and hence runs only authentic IP.
Complete design confidentiality	The legitimate client (this could be the system integrator, the end user, etc.) has only access to the design functionality as a black box (input/output behavior). No other party (in addition to the design developer) knows anything about the hardware IP.
Secure hardware IP updating	Given that there is already an authentic design running on the FPGA, the IP provider would like to update it and at a minimum keep all the security guarantees that the previous design kept.
Design traceability	Given an IP block, the designer can trace back who the intended recipient of the design was.
User privacy	A design should not be linkable to the identity of the end-user

6.2 Ultra-low cost anti-counterfeiting with LC-PUFs

We briefly describe how LC-PUFs can be applied as an anti-counterfeiting means. After an LC-PUF is created, it is embedded into the surface of a product, into packaging material or into a tamper evident seal that protects the packaging of multiple products. An enrollment measurement is done by performing a frequency sweep. Helper data W is generated from the response curve $Z(\omega)$, where ω denotes the frequency. Since there is nothing secret about the PUF characteristics, W may contain the full response curve in the clear. In practice it may be useful to include only a short representation of $Z(\omega)$. The helper data W further comprises the temperature T at enrollment. The enrollment data is either stored in a secure database or certified by a trusted party and stored next to the authentic product(s).

When a product has to be authenticated, the following steps are performed. First W is read. From W the verifier determines which frequency bands have to be investigated. A frequency sweep is done in those bands only, thus speeding up the verification. The temperature T' is measured. Finally, the measured response is compensated for the difference between T and T', and it is decided if the result is sufficiently close to the enrolled response.

For any commercial anti-counterfeiting technology it is important for the authenticity marks to be cheap. We estimate that with LCD manufacturing equipment, it is possible to bring the price of LC-PUFs to levels in which circuits for identification and anti-counterfeiting applications can become truly ambient.

6.3 Remote service/feature activation

Introduced in Guajardo et al. (2008a) and closely related to IP protection, remote service activation refers to the ability to enable certain features of a product once the product has been sold or is in possession of an external (and often) untrusted party. In this case, the aim is to allow only parties with the right credentials to be able to activate certain features of a product. Based on our discussion on fuzzy extractors in Sect. 3.3, if one is to reconstruct the key K based on a noisy response R', it is necessary to provide the procedure Rep with the helper data W. Thus, W can be used as a feature activation token even after the device is in the hands of an untrusted party. In addition, notice that thanks to the way in which the key K is derived no information about the key is leaked by the helper data W. Finally, W is specific to each PUF instance and, thus, to each device. In particular, the helper data W is specific to each device. Thus, enabling a feature after obtaining W_i for device i does not allow a user to activate the same feature for device j. We refer to Dodis et al. (2004); Linnartz and Tuyls (2003) for further discussions regarding security of different fuzzy extractor constructions.

6.4 Secret-key storage

A key observation in Tuyls et al. (2006) is that the coating can be used to store keys (rather than as a challenge-response repository as in previous works) and that these keys are not stored in memory. Rather, whenever an application requires the key, the key is generated on the fly. This makes it much more difficult for an attacker to compromise key material in security applications. Finally, Tuyls et al. (2006) show that active attacks on the coating can be easily detected, thus, making it a good countermeasure against probing attacks.

6.5 Authentication via challenge-response pairs

Challenge-response authentication techniques are based on the idea that a claimant or prover proves to a verifier knowledge of a secret without expressly revealing the secret. The authentication is performed with the help of a time varying value called the challenge usually chosen at random by the verifier. The response of the prover depends then on the challenge and on his/her secret value. Pappu (2001) was the first to propose using PUFs integrated into a CR protocol for authentication purposes. The basic idea is to go through an enrollment process (performed in a secure facility) in which a number of challenges and corresponding PUF responses are stored in a secure database. At a later stage, the prover, who wants to gain access to a service, contacts the verifier, who then sends the prover a challenge from the database, the prover challenges its PUF, records the PUF response and forwards it to the verifier. The verifier can then check if the response is the same one as the one stored in the database. If the check is positive, the verifier grants access to the requested service. Notice that this protocol assumes that each challenge is used once (otherwise replay attacks are possible). It is also assumed, as pointed out in Sect. 3, that without access to the right PUF, the probability of generating the expected response is negligible.

7 Conclusions

The promise of ambient intelligence will only achieve its true potential if we can guarantee that the information gathered around us is used in a privacy sensitive and secure manner. This, in turn, can only be achieved if we trust that the keys used to secure our sensitive information have not been compromised. In this paper, we have described how Physical Unclonable Functions and their corresponding Helper Data algorithm (or Fuzzy Extractor) can help us achieve these goals. In particular, we introduce two protocols for secure key deployment in the absence of any (previously) shared secret. Our protocols take advantage of specific fuzzy extractor properties to provide secrecy and authenticity of the deployed key against omni-present adversaries, i.e., adversaries that are present everywhere and all the time. Compared to previous protocols, and most prominently the Message-In-a-Bottle proposal (Kuo et al., 2007), our protocols are simpler (less communication complexity) and require less additional hardware. In addition, because of the use of PUFs, our solution provides tamper evidence and unclonability, valuable goals in themselves. Finally, we also introduce a new PUF construction aimed at ultra-low cost applications that need to guarantee their authenticity.

Bibliography

R. Anderson, H. Chan, and A. Perrig. Key Infection: Smart Trust for Smart Dust. In *IEEE International Conference on Network Protocols — ICNP 2004*, pages 206–215. IEEE Computer Society, October 5-8, 2004.

D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *Network and Distributed System Security Symposium — NDSS 2002*, 2002.

A. Bellaouar and M. I. Elmasry. *Low-Power Digital VLSI Design. Circuits and Systems.* Kluwer Academic Publishers, first edition, 1995.

M. Bellare and P. Rogaway. Entity Authentication and Key Distribution. In D. R. Stinson, editor, *Advances in Cryptology — CRYPTO '93*, volume 773 of *LNCS*, pages 232–249. Springer, August 22-26, 1993.

A. J. Bhavnagarwala, X. Tang, and J. D. Meindl. The Impact of Intrinsic Device Fluctuations on CMOS SRAM Cell Stability. *IEEE Journal of Solid-State Circuits*, 36(4):658–665, April 2001.

N. Bird, C. Conrado, J. Guajardo, S. Maubach, G.-J. Schrijen, B. Škorić, A. M. H. Tombeur, P. Thueringer, and P. Tuyls. ALGSICS - Combining Physics and Cryptography to Enhance Security and Privacy in RFID Systems. In F. Stajano, C. Meadows, S. Capkun, and T. Moore, editors, *Security and Privacy in Ad-hoc and Sensor Networks — ESAS 2007*, volume 4572 of *LNCS*, pages 187–202. Springer, July 2-3, 2007.

S. Bono, M. Green, A. Stubblefield, A. Juels, A. Rubin, and M. Szydlo. Security analysis of a cryptographically-enabled rfid device. In P. McDaniel, editor, *USENIX Security Symposium — Security '05*, pages 1–16, 2005.

X. Boyen. Reusable cryptographic fuzzy extractors. In V. Atluri, B. Pfitzmann, and P. D. McDaniel, editors, *ACM Conference on Computer and Communications Security — ACM CCS 2004*, pages 82–91. ACM, October 25-29, 2004.

X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In R. Cramer, editor, *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *LNCS*, pages 147–163. Springer-Verlag, 2005.

M. Cagalj, S. Capkun, and J. Hubaux. Key agreement in peer-to-peer wireless networks. *Proceedings of the IEEE (Special Issue on Cryptography and Security)*, 94(2):467–478, 2006.

D. Carluccio, K. Lemke, and C. Paar. E-passport: the global traceability or how to feel like an ups package. Printed handout of Workshop on RFID Security – RFIDSec 06, pages 167–180. ECRYPT Network of Excellence, July 2006 a. Available at `http://events.iaik.tugraz.at/RFIDSec06/Program/index.htm`.

D. Carluccio, T. Kasper, and C. Paar. Implementation details of a multi purpose ISO 14443 RFID-tool. Printed handout of Workshop on RFID Security – RFIDSec 06, pages 181–197. ECRYPT Network of Excellence, July 2006 b. Available at `http://events.iaik.tugraz.at/RFIDSec06/Program/index.htm`.

L. Carter and M. N. Wegman. Universal Classes of Hash Functions. *J. Comput. Syst. Sci.*, 18 (2):143–154, 1979.

C. Castelluccia and A. Francillon. TinyRNG, A Cryptographic Random Number Generator for Wireless Sensor Network Nodes. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks — IEEE WiOpt 2007*. IEEE, April, 2007.

C. Castelluccia and P. Mutaf. Shake them up!: a movement-based pairing protocol for CPU-constrained devices. In K. G. Shin, D. Kotz, and B. D. Noble, editors, *International Con-*

ference on Mobile Systems, Applications, and Services — MobiSys '05, pages 51–64. ACM, 2005.

H. Chan and A. Perrig. Security and privacy in sensor networks. *IEEE Computer*, 36(10): 103–105, 2003.

H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In *IEEE Symposium on Security and Privacy — S&P 2003*, pages 197–215. IEEE Computer Society, 2003.

B. Cheng, S. Roy, and A. Asenov. The impact of random doping effects on CMOS SRAM cell. In *European Solid State Circuits Conference*, pages 219–222, Washington, DC, USA, 2004. IEEE Computer Society.

G. DeJean and D. Kirovski. Making RFIDs unique - radio frequency certificates of authenticity. In *IEEE Antennas and Propagation Society International Symposium*, pages 1039–1042. IEEE, July 9-14, 2006.

J. Deng, C. Hartung, R. Han, and S. Mishra. A practical study of transitory master key establishment forwireless sensor networks. In *International Conference on Security and Privacy for Emerging Areas in Communications Networks — SECURECOMM'05*, pages 289–302, Washington, DC, USA, 2005. IEEE Computer Society.

Y. Dodis, M. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology —- EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer-Verlag, 2004.

W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In S. Jajodia, V. Atluri, and T. Jaeger, editors, *ACM Conference on Computer and Communications Security — CCS 2003*, pages 42–51, New York, NY, USA, 2003. ACM.

J. Eagle. RFID: The Early Years 1980-1990. Available at `http://members.surfbest.net/eaglesnest/rfidhist.htm`, 2002. Website. Updated September 27th, 2002.

D. W. Engels and S. Sarma. Standardization Requirements within the RFID Class Structure Framework. Technical report, Auto-ID Laboratories, Massachusetts Institute of Technology, Cambridge, MA 02139-4307, USA, January 2005. Available at `http://ken.mit.edu/web/`.

L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In V. Atluri, editor, *ACM Conference on Computer and Communications Security — CCS 2002*, pages 41–47, New York, NY, USA, 2002. ACM.

B. Gassend. Physical Random Functions. Master's thesis, Computer Science and Artificial Intelligence Laboratory, MIT, February 2003. Computation Structures Group Memo 458.

B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Controlled Physical Random Functions. In *Annual Computer Security Applications Conference — ACSAC 2002*, page 149, Washington, DC, USA, 2002a. IEEE Computer Society. ISBN 0-7695-1828-1.

B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas. Silicon physical unknown functions. In V. Atluri, editor, *ACM Conference on Computer and Communications Security — CCS 2002*, pages 148–160. ACM, November 2002b.

J. Guajardo, R. Blümel, U. Krieger, and C. Paar. Efficient Implementation of Elliptic Curve Cryptosystems on the TI MSP 430x33x Family of Microcontrollers. In K. Kim, editor, *International Workshop on Practice and Theory in Public Key Cryptography — PKC 2001*, volume 1992 of *LNCS*, pages 365–382. Springer, February 13-15, 2001.

J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls. Physical Unclonable Functions and Public Key Crypto for FPGA IP Protection. In *International Conference on Field Programmable Logic and Applications — FPL 2007*, pages 189–195. IEEE, August 27-30, 2007a.

J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls. FPGA Intrinsic PUFs and Their Use for IP Protection. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and*

Embedded Systems — CHES 2007, volume 4727 of *LNCS*, pages 63–80. Springer, September 10-13, 2007b.

J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls. Brand and IP Protection with Physical Unclonable Functions. In *IEEE International Symposium on Circuits and Systems — ISCAS 2008*, pages 3186–3189. IEEE, May 18-21, 2008a.

J. Guajardo, P. Tuyls, N. Bird, C. Conrado, S. Maubach, G.-J. Schrijen, B. Škorić, A. Tombeur, and P. Thueringer. RFID Security: Cryptography and Physics Perspectives. In P. Kitsos and Y. Zhang, editors, *RFID Security: Techniques, Protocols and System-On-Chip Design.* Springer, 2008b. To appear.

D. E. Holcomb, W. P. Burleson, and K. Fu. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. Conference on RFID Security 07, July 11-13, 2007.

L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Ubicomp 2001: Ubiquitous Computing, Third International Conference*, pages 116–122, 2001.

V. Hsu, J. M. Kahn, and K. S. J. Pister. Wireless Communications for Smart Dust. Electronics Research Laboratory Technical Memorandum Number M98/2, University California Berkeley, 1998.

ICC. ICC Policy Statement: The fight against piracy and counterfeiting of intellectual property. Submitted to the 35th World Congress, Marrakech, Document no 450/986, International Chamber of Commerce, June 1st, 2004.

A. Juels. RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, February 2006. Extended version available from `http://www.rsasecurity.com/rsalabs/node.asp?id=2029`.

A. Juels and M. Wattenberg. A Fuzzy Commitment Scheme. In J. Motiwalla and G. Tsudik, editors, *ACM Conference on Computer and Communications Security — ACM CCS '99*, pages 28–36. ACM, November 1-4, 1999.

A. Juels, R. Pappu, and S. Garfinkel. RFID Privacy: An Overview of Problems and Proposed Solutions. *IEEE Security and Privacy*, 3(3):34–43, May/June 2005. Extended version available from `http://www.rsasecurity.com/rsalabs/node.asp?id=2029`.

A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe. Watermarking techniques for intellectual property protection. In *Design Automation Conference — DAC '98*, pages 776–781, New York, NY, USA, 1998. ACM Press.

J.-P. Kaps, K. Y., and B. Sunar. Energy Scalable Universal Hashing. *IEEE Trans. Computers*, 54(12):1484–1495, 2005.

T. Kean. Cryptographic rights management of FPGA intellectual property cores. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays — FPGA 2002*, pages 113–118, 2002.

H. Krawczyk. LFSR-based Hashing and Authentication. In Y. Desmedt, editor, *Advances in Cryptology - CRYPTO '94*, volume 839 of *LNCS*, pages 129–139. Springer, August 21-25, 1994.

C. Kuo, M. Luk, R. Negi, and A. Perrig. Message-In-a-Bottle: User-Friendly and Secure Key Deployment for Sensor Nodes. In *International Conference on Embedded Networked Sensor Systems — SenSys '07*, pages 233–246. ACM, 2007.

M. Lacey. Panama: Tainted Syrup Now Linked to Deaths. The New York Times. Available at `http://www.nytimes.com`, October 13, 2006. World Briefing — Americas.

J. Landt. Shrouds of Time — The History of RFID. Whitepaper, AIM Inc., October 1st, 2001. Available at `http://www.transcore.com/pdf/AIM%20shrouds_of_time.pdf`.

J. Lester, B. Hannaford, and G. Borriello. ”are you with me?” - using accelerometers to determine if two devices are carried by the same person. In *Pervasive Computing, Second International Conference*, pages 33–50, 2004.

D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1200–1205, October 2005. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1561249`.

J.-P. M. G. Linnartz and P. Tuyls. New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates. In J. Kittler and M. S. Nixon, editors, *Audio-and Video-Based Biometrie Person Authentication — AVBPA 2003*, volume 2688 of *LNCS*, pages 393–402. Springer, June 9-11, 2003.

D. Liu, P. Ning, and W. Du. Group-based key pre-distribution in wireless sensor networks. In M. Jakobsson and R. Poovendran, editors, *ACM Workshop on Wireless Security — WiSe 2005*, pages 11–20, New York, NY, USA, 2005. ACM.

K. Lorincz, D. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, S. Moulton, and M. Welsh. Sensor Networks for Emergency Response: Challenges and Opportunities. *IEEE Pervasive Computing, Special Issue on Pervasive Computing for First Response*, 3:16–23, Oct-Dec 2004.

J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication. In *IEEE Symposium on Security and Privacy — S&P 2005*, pages 110–124. IEEE Computer Society, May 8-11, 2005.

A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

W. Nevelsteen and B. Preneel. Software Performance of Universal Hash Functions. In J. Stern, editor, *Advances in Cryptology — EUROCRYPT'99*, volume 1592 of *LNCS*, pages 24–41. Springer, May 2-6, 1999.

C. O'Donnel, G. Suh, and S. Devadas. PUF-Based Random Number Generation. Technical Memo MIT-CSAIL-CSG-481, MIT CSAIL, November 2004.

Y. Oren and A. Shamir. Power Analysis of RFID Tags. Original announcement at RSA Conference 2006, February 14th, 2006. Webpage available at `http://www.wisdom.weizmann.ac.il/~yossio/rfid/`.

R. S. Pappu. *Physical one-way functions*. PhD thesis, Massachusetts Institute of Technology, March 2001. Available at `http://pubs.media.mit.edu/pubs/papers/01.03.pappuphd.powf.pdf`.

R. S. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297 (6):2026–2030, 2002. Available at `http://web.media.mit.edu/~brecht/papers/02.PapEA.powf.pdf`.

A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. SPINS: Security Protocols for Sensor Networks. *Wireless Networks*, 8(5):521–534, 2002.

A. Perrig, J. A. Stankovic, and D. Wagner. Security in wireless sensor networks. *Communications of the ACM*, 47(6):53–57, 2004.

B. Przydatek, D. X. Song, and A. Perrig. SIA: secure information aggregation in sensor networks. In I. F. Akyildiz, D. Estrin, D. E. Culler, and M. B. Srivastava, editors, *International Conference on Embedded Networked Sensor Systems — SenSys 2003*, pages 255–265. ACM, November 5-7, 2003.

M. Ramkumar and N. Memon. An efficient key predistribution scheme for ad hoc network security. *IEEE Journal on Selected Areas in Communications*, 23(3):611–621, 2005.

K. B. Rasmussen and S. Capkun. Implications of radio fingerprinting on the security of sensor networks. In *International Conference on Security and Privacy in Communication Networkds — SecureComm 2007*. IEEE, September 17-20, 2007.

S. Sarma and D. W. Engels. On the Future of RFID Tags and Protocols. Technical report mit-autoid-tr-018, Auto-ID Center, Massachusetts Institute of Technology, Cambridge, MA 02139-4307, USA, June 1st, 2003. Early Released July 2003. Available at `http://www.epcglobalinc.org/standards_technology/specifications.html`.

E. Seevinck, F. J. List, and J. Lohstroh. Static-Noise Margin Analysis of MOS SRAM Cells. *IEEE Journal of Solid-State Circuits*, 22(5):748–754, Oct 1987.

V. Shnayder, B. Chen, K. Lorincz, T. R. F. Fulford-Jones, and M. Welsh. Sensor networks for medical care. In J. Redi, H. Balakrishnan, and F. Zhao, editors, *International Conference on Embedded Networked Sensor Systems — SenSys 2005*, page 314. ACM, November 2-4, 2005.

V. Shoup. On Fast and Provably Secure Message Authentication Based on Universal Hashing. In N. Koblitz, editor, *Advances in Cryptology - CRYPTO '96*, volume 1109 of *LNCS*, pages 313–328. Springer, August 18-22, 1996.

E. Simpson and P. Schaumont. Offline Hardware/Software Authentication for Reconfigurable Platforms. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems — CHES 2006*, volume 4249 of *LNCS*, pages 311–323. Springer, October 10-13, 2006.

T. Staake, F. Thiesse, and E. Fleisch. Extending the EPC Network – The Potential of RFID in Anti-Counterfeiting. In A. O. H. Haddad, L. M. Liebrock and R. L. Wainwright, editors, *ACM Symposium on Applied Computing — SAC 2005*, pages 1607–1612. ACM Press, March 13-17 2005.

F. Stajano. The Resurrecting Duckling - What Next? In B. Christianson, B. Crispo, and M. Roe, editors, *Security Protocols Workshop. Revised Papers*, volume 2133 of *LNCS*, pages 204–214. Springer, April 3-5, 2000.

F. Stajano and R. J. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, editors, *Security Protocols*, volume 1796 of *LNCS*, pages 172–182. Springer-Verlag, April 19-21, 1999.

Y. Su, J. Holleman, and B. Otis. A 1.6pJ/bit 96% Stable Chip-ID Generating Cicuit using Process Variations. In *ISSCC '07: IEEE International Solid-State Circuits Conference*, pages 406–408, Washington, DC, USA, 2007. IEEE Computer Society.

P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters. Read-Proof Hardware from Protective Coatings. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems — CHES 2006*, volume 4249 of *LNCS*, pages 369–383. Springer, October 10-13, 2006.

B. Škorić, P. Tuyls, and W. Ophey. Robust Key Extraction from Physical Uncloneable Functions. In J. Ioannidis, A. D. Keromytis, and M. Yung, editors, *Applied Cryptography and Network Security — ACNS 2005*, volume 3531 of *LNCS*, pages 407–422, June 7-10, 2005.

B. Škorić, G.-J. Schirjen, W. Ophey, R. Wolters, N. Verhaegh, and J. v. Geloven. Experimental Hardware for Coating PUFs and Optical PUFs. In P. Tuyls, B. Škorić, and T. Kevenaar, editors, *Security with Noisy Data*, pages 255–268. Springer-Verlag, first edition, 2007.

M. Weiser. The Computer for the Twenty-First Century. *Scientific American Magazine*, 265 (3):94–100, September 1991.

G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a Wireless Sensor Network on an Active Volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

J. L. Wong, J. Feng, D. Kirovski, and M. Potkonjak. Security in sensor networks: watermarking techniques. In C. S. Raghavendra, K. M. Sivalingam, and T. Znati, editors, *Wireless sensor networks*, pages 305–323. Kluwer Academic Publishers, 2004.

S. Zhu, S. Setia, and S. Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In S. Jajodia, V. Atluri, and T. Jaeger, editors, *ACM Conference on Com-*

puter and Communications Security — CCS 2003, pages 62–72, New York, NY, USA, 2003. ACM. ISBN 1-58113-738-9.

S. Zhu, S. Setia, and S. Jajodia. Leap+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks*, 2(4):500–528, 2006.

Zigbee Specification. ZigBee Alliance, June 2005. Document 053474r06, Version 1.0.

Jorge Guajardo is a senior scientist in the Information and System Security Department at Philips Research Europe. There he lead the efforts to design new and efficient methodologies to secure RFID systems and since 2007 has focus on the design of new anti-counterfeiting methodologies based on Physical Unclonable Functions (PUFs) and their applications to secure key storage and wireless sensor networks. Previous to joining Philips Research, Jorge worked for GTE Government Systems, RSA Laboratories, cv cryptovision gmbh, and Infineon Technologies AG. His interests include: the efficient implementation of cryptographic algorithms in constrained environments, the development of hardware architectures for private and public-key algorithms, provable security of cryptographic protocols under various assumptions, and the interplay of physics and cryptography to attain security goals. Jorge holds a B.Sc degree in physics and electrical engineering and M.S. in electrical engineering from Worcester Polytechnic Institute and a Ph.D. degree in electrical engineering and information sciences from the Ruhr-Universitaet Bochum obtained under the supervision of Prof. Christof Paar.

Boris Škorić received a PhD in theoretical physics from the University of Amsterdam, the Netherlands, in 1999. From 1999 to 2008 he was a research scientist at Philips Research in Eindhoven, working first on display physics and later on security topics. In 2008 he joined the faculty of Mathematics and Computer Science of Eindhoven Technical University, the Netherlands, as assistant professor.

Pim Tuyls studied Theoretical Physics at the Katholieke Universiteit of Leuven where he got a Ph.D. on Quantum Dynamical Entropy in 1997. Currently he works as Chief Technologist at Philips Intrinsic ID in the Netherlands where he is leading the crypto development activities. Since 2004, he is also a visiting professor at the Cosic institute in Leuven. His main interests are in Key Extraction from Noisy Data (Physical Unclonable Functions and Private Biometrics, Quantum Cryptography) and in applications of Secure Multi-Party Computation.

Sandeep S. Kumar is a Senior Researcher at Philips Research Europe. Kumar received both his B.Tech. and M.Tech. degrees in Electrical Engineering from IIT-Bombay, India in 2002. He received his Ph.D. degree in Communication Security from Ruhr University Bochum, Germany in 2006. His research interests include hardware and software architectures for implementations of cryptographic systems, in particular elliptic-curve cryptography on constrained devices. At Philips Research he has been working on hardware implementations of physically unclonable functions for anti-counterfeiting and presently on identity management systems for lifestyle applications. He is a member of the IACR.

Thijs Bel studied Chemical Differentation at the IHBO of Eindhoven. He obtained his certificate in 1984. In 1985 he joined Philips Research, first working on lithography for IC's and later on lithography for several kinds of displays. In 2007 he joined the group Thin Film Facilities, where he has been working on PUFs and in 2008 he joined the group Device processing Facilities, working on OLEDs.

Antoon H. M. Blom studied electro technology at the Technical High School of s Hertogenbosch, where he graduated in 1978. In 1979 he joined the Philips Company at the mechanization department of the Volt site in Tilburg, a production site for wire wound components. After an intermediate period at the laboratory for tuning units and transformers within the consumer electronics department in Eindhoven, he joined the centre for manufacturing technologies, which has recently been merged with the Philips Applied Technologies department, where he is working in the Optics & Sensors group of the Process Technology department.

Geert-Jan Schrijen obtained his M.Sc. degree in Electrical Engineering from the University of Twente (Enschede) in December 2000. During his studies he specialized in digital signal processing and active noise cancellation. In April 2001 he joined Philips Research. As a research scientist he became interested in the fields of cryptography and information theory and worked several years on security technologies like Digital Rights Management (DRM) systems, low-power authentication protocols and private biometric systems. From 2005 he has been involved in the work on Physical Unclonable Functions (PUFs). Geert-Jan was appointed Chief Algorithm Development at the Philips Intrinsic-ID lab venture in April 2007, where he is focusing on the development of signal processing algorithms and security architectures around PUFs.