Using Physical Unclonable Functions for Hardware Authentication: A Survey

Susana Eiroa, Iluminada Baturone, Antonio J. Acosta
Depto. Electrónica y Electromagnetismo, Univ. de Sevilla
IMSE-CNM-CSIC
Seville, Spain
{eiroa, lumi, acojim}@imse-cnm.csic.es

Jorge Dávila
Depto. Lenguajes y Sist. Informáticos e Ing. del Software
Univ. Politécnica de Madrid
Madrid, Spain
jdavila@fi.upm.es

Abstract— **Physical unclonable functions (PUFs) are drawing a crescent interest in hardware oriented security due to their special characteristics of simplicity and safety. However, their nature as well as early stage of study makes them constitute currently a diverse and non-standardized set for designers. This work tries to establish one organization of existing PUF structures, giving guidelines for their choice, conditioning, and adaptation depending on the target application. In particular, it is described how using PUFs adequately could enlighten significantly most of the security primitives, making them very suitable for authenticating constrained resource platforms.**

Keywords-PUFs; hardware security; light cryptography

I. Introduction

Microelectronics and Telecommunications have undertaken an incredible evolution during last decades. This together with the social trend of globalization and so ubiquity, derived in the amazing development of integrated circuits (ICs) for multiple applications. In parallel, the market of the Intellectual Property (IP) is growing in a spectacular manner due to the evolution of FPGAs. Both, ICs and FPGAs, include each time more capabilities among which we can find security features. Security issues range from IC identification and IP protection in FPGAs to trusted computing and cryptographic aspects. Regarding the latter, several tasks are required, such as secret-key storage, key distribution or key generation, including secure protocols to authenticate communication extremes as well as messages, which is particularly relevant in the case of hardware tokens for individual authentication.

From their introduction by Pappu in 2001 [1], security community has paid great attention to PUFs (Physical Unclonable Functions or Physical Random Functions). They have been considered as a key aspect in security for their unclonability, randomness, and resiliency against physical attacks (e.g., radiation and reverse engineering). The original proposal in [1] was based on the scattering obtained when shining a laser on a bubble-filled transparent epoxy wafer. Later on, Gassend *et al.* introduced silicon PUFs [2], which exploit manufacturing process variations in ICs, that is, random variables of the fabrication process, which during many years were seen as a problem for circuit design, are now utilized for security purposes. Process variations introduced during the manufacturing process are beyond the control of manufacturers and increase with submicron technologies. Hence, silicon PUFs can be obtained without any special manufacturing steps. In addition, their simple structures introduce much lower time, speed, and power overheads than other cryptography-based security techniques.

The problem is that current PUF-based applications constitute a heterogeneous space where most of the systems are designed ‘ad-hoc’. Consequently, someone wishing to understand this field may be confused by the variety of PUFs reported and the different scenarios where they can be employed.

This paper summarizes the use of PUFs for hardware authentication. Section II is dedicated to describe briefly what a PUF is and how it can be used for authentication purposes. Section III shows the PUF structures reported in the literature classified accordingly to its working principle. Section IV describes configurations to enhance their security while Section V reviews basic structures based on PUFs for IC identification, random number generation, secret key generation, and physical obfuscating a secret. Section VI illustrates how these primitives can be exploited to implement secure authentication protocols, in particular, in a low-cost protocol, which is adequate for security hardware tokens. Finally some conclusions are given in Section VII.

II. Physical Unclonable Functions

A PUF is a Physical Random Function that maps a set of challenges to a set of responses driven by parametric properties of physical components that are difficult to predict, control, or reproduce. Therefore, the mapping function can only be evaluated with the physical system, and it is unique for each physical instance. Pappu [1] defined a PUF as a physical object with the following properties:

- It can be subjected to a large number of different challenges that yield unpredictable responses.
- It is very hard to clone physically.
- Mathematical modeling of the challenge-response pairs is very difficult.

This work was partially supported by Junta de Andalucía under the Project P08-TIC-03674, by the European Community through the MOBY-DIC Project FP7-INFSO-ICT-248858 (www.mobydic-project.eu), and by Spanish Ministerio de Ciencia y Tecnología under the Projects TEC2008-04920, DPI2008-03847, and TEC2007-65105.

- It is hard to characterize its physical structure.

The first property aims at avoiding statistical attacks based on prediction and replay attacks that look for repeated challenges. Regarding this issue, predictability tests (which measure relationships between challenge-response pairs) are very interesting for PUFs [3]. The other properties reduce the problem of model-building attacks, that is, attackers find it difficult to construct a software or hardware model of a PUF from evaluating challenge-response pairs. Anyway, model-building attacks are not a concern whenever the challenges or the responses are not exposed, which happens in several application scenarios.

In order to use PUFs in authentication systems, they should fulfill the following statistical properties:

- Inter-class Hamming variation should be ideally of μ=50% with a typical deviation of σ=0%. The Hamming inter-class distance represents the difference between two responses of different PUFs to the same challenge. This property measures the PUF uniqueness, that is, how distinctly the PUF can identify the circuit where it is included.
- Intra-class Hamming variation should be ideally of μ=0% with a typical deviation of σ=0%. The Hamming intra-class distance represents the difference between two responses of one PUF to the same challenge. This property measures the PUF reliability, that is, how consistently a response is reproduced by the PUF for the same challenge over several read outs.

These properties should be verified ideally independently of aging and environmental conditions (varying temperature, fluctuating supply voltage, and so on). Hence, sensitivity tests of the PUFs are very important [3].

III. CLASSIFICATION OF PUFS

First step in the selection of a PUF is to know the alternatives together with the arguments for and against each of them. Next classification firstly divides PUFs into two different groups: those that can only be implemented in ICs with special fabrication steps, and those that can be implemented in both FPGAs and ICs of standard manufacturing processes. Within each group, the types of PUFs are clustered accordingly to the manufacturing variability parameter that determines their working mode.

A. PUFs requiring special fabrication process

They are based on the measurements taken from one special layer of deposed material that contains particular embedded elements. The type of elements and the measurement mode establish the difference between them.

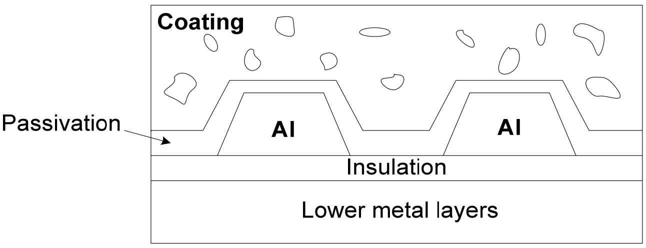

Figure 1. Transversal section of a coating PUF [4]

- *Coating PUFs* [4]: One layer of dielectric material containing dielectric particles of different permittivity is deposed over an IC (Fig. 1). Between the coating layer and the rest of the IC an array of sensors is placed to measure the capacitance of the layer (response). This way the measurement circuit is protected by the layer from inspection. Coating PUFs have a small number of challenge-response pairs because the amount of sensors is limited and every sensor yields only a few bits.
- *Optical PUFs* [5]: The PUF is formed by a piece of glass that contains light scattering particles. If it is irradiated with, e.g. a laser beam, different patterns (responses) are obtained. The physical structure can be optionally integrated with the laser and the reading device into an IC. A disadvantage of these PUFs is that it is difficult to evaluate their uniqueness and ensure a good reliability.

B. Silicon PUFs

The working principle of silicon PUFs [2] is exploitation of small variations in the IC manufacturing process. The parameter selected will define the resulting type of PUF.

- *Leakage current based PUFs* [6]: They are based on the idea that different physical realizations of the same circuit have different leakage current consumption. Their measure and binarization provide different digital responses for each selected circuit.
- *Delay based PUFs***:** Even using identical layout masks, manufacturing variability cause different delays in different realizations of the same circuit. The main constructions are:
  - *Arbiter PUFs*: They contain two paths with the same layout length that are selected by a multiple-bit input (challenge) via multiplexors (Fig. 2a). Given a rising signal to both paths at the same time, the signals race through them until an arbiter (latch) at the end decides which one is faster, thus giving a different output (response) [7]. There is one variant to this scheme called *tristate buffer* [8] where the multiplexors are replaced by pairs of tristate buffers. Since arbiter PUFs are very simple, they are very suitable for resource-constrained applications. As a drawback, attackers can build a precise timing model of them from challenge-response pairs.
  - *Ring oscillator PUFs*: They are formed by a set of identical ring oscillators that oscillate at slightly different frequencies due to variations in the fabrication process (Fig. 2.b) [7]. Advantages of these PUFs compared with arbiter ones are an

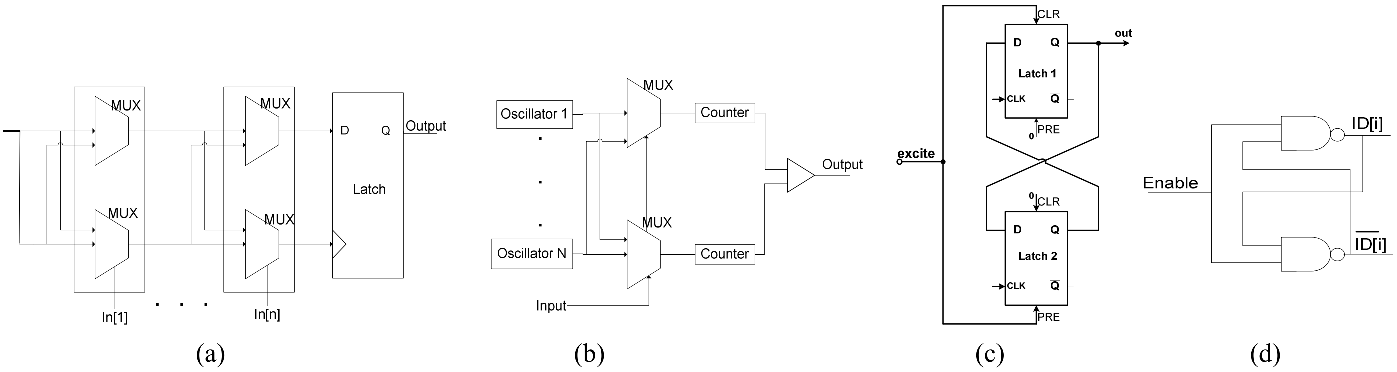

Figure 2. Structures of (a) Arbiter PUF, (b) Ring oscillator PUF, (c) Butterfly PUF, and (d) NOR-based PUF

easier implementation for both ASICs and FPGAs, easier evaluation of entropy, and higher uniqueness and reliability. As disadvantages, they are slower, larger, and consume more power than arbiter PUFs.

- *Memory based PUFs*: They are based on cross-coupled circuits that have two different stable operating points and one unstable point. If the circuit is not driven by any input, the slight differences (mismatching) between the two ideally symmetrical parts motivate the circuit goes more often to one of the stable states. Every memory element yields one response bit, so that a challenge-response behavior can be mimicked by using more memory cells than response bits. The drawback is that the hardware resources scale exponentially with the challenge size. The advantage is that they are superior to delay PUFs in speed and power consumption.
    - *SRAM PUFs*: They were the first practical PUFs devoted to FPGAs since they are intrinsic in many FPGAs [9]-[10]. A disadvantage is that an attacker could read initial value of every memory location in the device. Another disadvantage is that not all the FPGAs support uninitialized SRAM memory.
    - *Butterfly and NOR-based PUFs*: They are based on cross-coupled latches [11] (Fig. 2c) or NOR gates (Fig. 2d) [12]. They are not intrinsic PUFs but can be implemented in standard FPGAs or ASICs.

Table 1 summarizes the measurements that have been reported for the different PUFs reviewed in this section. Current research is being conducted towards improving reliability and reducing sensitivity [13]. This is why the measurements shown for Ring Oscillator PUFs are so good.

TABLE I. MEASUREMENTS REPORTED OF DIFFERENT PUFS

PUF Type	**PUF Quality Factors**					
	Inter-class Hamming Distance	Intra-class Hamming Distance	Sensitivity to environment			Implementation Difficulty
			Ageing	Temperature	Voltage	
Coating [4]	μ=50%	μ=4.4%	-	Very high	-	High
Optical [5]	Impractical to measure	Very high	-	Very high	-	Very high
Leakage Current [6] (180 nm)	μ=50%	μ=2%	-	Very high	Very high	Medium
Arbiter [7] (180 nm)	μ=23%	μ=0.7%	0.7% Intra-class	4.82% (20 to 70ºC)	3.74% (2% ΔV)	Low
Ring Oscillator [13] (90 nm)	μ=45.51%	μ=0%	-	0% (25 to 65ºC)	3.15% (20% ΔV)	Very low
SRAM [9]-[10] (90 nm)	μ=49.97% σ=0.3%	μ=3.57% σ=0.13%	4.5% Intra-class	12% (-20 to 80ºC)	-	Very low
Butterfly [11] (65 nm)	μ=50%	μ=6%	-	2.3% (-20 to 80ºC)	-	Very low
NOR [12] (130 nm)	μ=50.13% σ=0.6%	μ=3.89% σ=0.21%	4% Intra-class	3.91% (0 to 70ºC)	5.47% (20% ΔV)	Low

IV. Enhancing Security of PUFs

Some constructions using PUFs have been proposed in order to improve security parameters. One of them is the concept of Reconfigurable PUF (RPUF) proposed by Lim in [14]. Another is the concept of Controlled PUF (CPUF), which was introduced by Gassend *et al.* in [15].

A. *Reconfigurable PUFs (RPUFs)*

Opposite to non reconfigurable PUFs, RPUFs show a dynamic performance that is obtained through an ideally unpredictable mechanism that alters the challenge-response behavior of the PUF and produces a new PUF that inherits all the security properties of the original.

The approach reported in [5] to obtain a reconfigurable optical PUF consists in modifying the PUF structure. The reconfiguration process is driven by a laser with high intensity that makes the glass material of the PUF melt. After cooling the structure, light scattering particles experiment reorientations, thus providing different challenge-response behavior in the moment of being irradiated.

Another approach reported in [16] is to use two reconfigurable networks with a group of PUFs in parallel. An input network that makes one mapping of the input challenges to the PUFs, and an output network that combines the output of the parallel PUFs to obtain the output of the system. The PUF is reconfigured by changing the mapping and combining functions.

B. *Controlled PUFs (CPUFs)*

A CPUF is defined in [15] as a PUF combined with a general-purpose processing element that controls the access to the PUF through a specific application programming interface (Fig. 3). The processing element protects PUF against man-in-the-middle attacks because only authorized users by the control algorithm have access to the PUF's inputs and outputs. In the meantime, the PUF provides anti-tampering protection. The interface proposed in [15] for limiting access to PUFs can be applied to different scenarios.

V. Security Primitives Using PUFs

This section summarizes four main primitives that use PUFs for hardware authentication: (i) creation of identification number (ID), (ii) random number generation, (iii) secret key generation, and (iv) physical obfuscation of secrets.

A. *PUFs for ID creation*

Uniquely identifying ICs by an identification number (ID) is important for labeling RFID tags, addressing resource-constrained wireless sensor nodes, controlling the quality of IC fabrication process, or tracking implantable electronic devices. Advantages of using PUFs for ID creation are that the ID number is difficult to counterfeit, which is very important because counterfeiting is a current huge problem that not only produces economic losses but also threatens safety and health. The PUFs used in this application scenario must generate a digital output to form a binary ID code, and the output must be reliable over environmental changes.

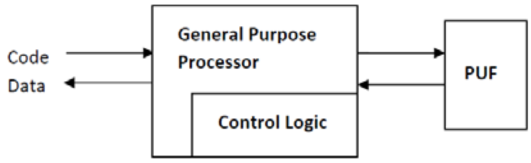

Figure 3. Block diagram of a CPUF

A sort of leakage current based PUFs are employed in [6] for this purpose. The leakage current value of a cell is continuous or analog, and so, the key point in this work is the conversion of the analog value into a digital one. Such conversion is done in two steps. Firstly, the model of the cell leakage is obtained and then these values are codified to obtain the ID number.

NOR based PUFs are employed in [12]. The ID is directly obtained by combining the output bits of several basic cells as the one shown in Fig. 2d. This solution is very efficient in terms of readout speed and power consumption. In order to increase the reliability of PUF output, they suggest reducing the influence of unstable bits by: (a) averaging multiple reads to attenuate the effects of thermal noise, (b) increasing the ID code length, or (c) increasing the signal-to-noise ratio of the circuit evaluation process.

B. *PUFs for random number generation*

Random numbers are the basis of most of the primitives and algorithms used in the security world. Approaches to creating random numbers can be classified into two categories: True Random Number Generation (TRNG) and Pseudo Random Number Generation (PRNG). PRNGs produce numbers that are random from a statistical point of view, although they are fully deterministic. TRNGs, which rely on a physical random process as a source of entropy, are desirable for security applications.

O'Donnell *et al.* proposed in [17] the use of silicon PUFs (arbiter PUFs) to generate random numbers. The underlying idea is to use challenges that return inconsistent responses, that is, the opposite to what is desired for achieving PUF reliability. These meta-stable challenges generate responses that can vary unpredictably. Attention should be paid to environmental changes because they are much more influential to this meta-stable behavior. Ring-oscillator [18] and memory based PUFs [19]-[20] have also been used for random number generation.

C. *PUFs for secret key generation*

PUFs are a good option for secret key generation as they provide security against Side Channel Attacks. In this application scenario, the reliability of the PUF is very important because the response provided to the same challenge should always be the same. Helper Data or Fuzzy Extractor Algorithms have been employed to cope with noisy PUF responses [21]. They consist of the following phases:

- Enrollment: A helper data vector, W, is generated from the PUF response, R, added (usually XORed) to one codeword *c*, which is chosen randomly from an

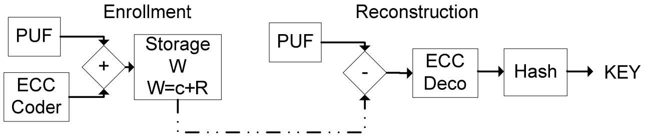

Figure 4. Cryptographic key generation with PUFs

error correction code (ECC) C. The helper data can be made public while the codeword must be private.

- Key reconstruction: This phase consists of two steps named 'information reconciliation' and 'privacy amplification'. In the first step, the PUF is challenged as in the enrollment, now obtaining a probably noisy response, R'. This output is subtracted to the helper data (usually XOR(W,R') is used) and the result goes through the ECC decoding to recover the codeword *c*. From c, the original PUF response, R, can be reconstructed (R = XOR(W,*c*)). In the 'privacy amplification' step, a hash function is used to generate the key so as to provide well randomness.

An example of secret key generation scheme using PUFs is given in Fig. 4.

D. PUFs for physical obfuscating a secret

Gassend *et al.* in [22] propose a scheme called physical obfuscated key (POK) for achieving secret key storage. The underling idea of POK is to split the computation of the key into two steps and relate one of them to a PUF. For example, the key is obtained by combining (via an XOR) the output of the PUF with some fuse or any data stored in an EEPROM. The complete key is not stored permanently but it is accessible only after the PUF is stimulated. Once erased from volatile memory, the value of the key is no longer available. In this scenario, a fixed hard-wired challenge is applied to the PUF so as to make the PUF response ideally the same. Hence, the other data combined with the PUF should be changed in order to change the key.

Tuyls *et al.* in [23] describes another scheme based on PUFs for secret key storage. A long-term key is stored in a non-secure memory. The short-term key used for encrypting the long-term key is extracted from the PUF. In this scenario, many challenges are applied to the PUF so as to generate different short-term keys every time the PUF is used.

Bringer *et al.* in [24] propose extending the use of physical obfuscation to binary operations such as XOR, AND, and scalar product, and apply them for obfuscating the secret material of linear as well as non linear stream ciphers.

VI. PUFs for Secure Authentication Protocols

Several protocols have been proposed that use PUFs in order to provide security features. Most of them are 'ad-hoc' structures whose construction depends on the application.

Guajardo *et al.* proposed in [10] a symmetric key protocol using PUFs as key generators. The construction targets to provide IP protection. In the meantime, it ensures confidentiality between the system where the IP would be implemented and the IP provider. Tuyls *et al.* also use PUFs as key generators in [25]. In this case, PUFs are building blocks in an off-line authentication protocol based on asymmetric cryptography. Since the target platforms are RFID systems with constrained resources, they employ elliptic curves that claim to be less resource consuming than other asymmetric approaches even they are more complex to develop.

Hammouri proposed in [26] a lightweight cryptography solution called HB-PUF protocol. It is based on merging two types of cryptographic primitives: PUFs and Hoper-Blum (HB) protocols [27]-[28]. The proposal of Kulseng in [29] also follows the idea of implementing minimalistic cryptography exploiting PUFs and LSFR registers. Both are symmetric key protocols.

Symmetric key constructions can always be simpler than asymmetric ones. In the field of security hardware tokens that require low cost implementation, PUFs can be a relevant constituent block not only to obstruct replicating the token but also to implement symmetric key constructions with low hardware resources. Resistance of symmetric key protocols to man-in-the-middle attacks (using PUFs or not) depends on the secrecy and security employed when sharing secrets in the enrollment phase.

Fig. 5 shows a possible scheme of a mutual (reader and token) authentication protocol using PUFs. Several of the structures explained above can be used to both enlighten and easily implement security features. In particular, they are used in the scheme shown in Fig. 5 to:

- Generate the random challenges (nonces) in each authentication part.
- Generate the secret key, K_{AB}.

VII. Conclusions

Selection of the correct PUF for one defined security application could become a tricky task due to the several parameters that should be considered. Since variations in the manufacturing process and the mapping function associated determine the PUF behavior, they should be selected accordingly to the implementation target (FPGA or IC) of the application and the quality factors required. Depending on the security primitive in which the PUF is involved, it might be necessary to implement especial algorithms like helper data or using other blocks such as fuses. Due to PUF characteristics of simplicity and safety, they can be used as relevant constituent blocks in authentication systems, especially in hardware tokens with constrained resources.

References

[1] R. Pappu. "Physical one-way functions", PhD Thesis, MIT, 2001.

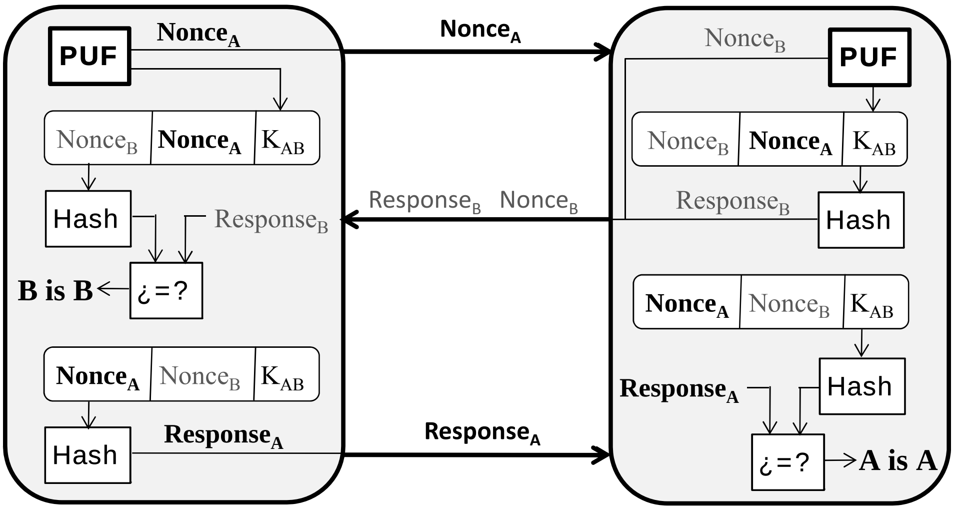

Figure 5. Challenge-response protocol scheme using PUFs

[2] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas. “Silicon physical unknown functions”, in Proc. ACM Conf. on Computer and Communications Security, CCS 2002, pp. 148–160.

[3] M. Majzoobi, F. Koushanfar, and M. Potkonjak. “Testing techniques for hardware security”, in Proc. IEEE Int. Test Conference, ITC 2008, pp. 1-10.

[4] P. Tuyls, G.-J. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh and R. Wolters: "Read-proof hardware from protective coatings", in Proc. of the 8th Int. Workshop on Cryptographic Hardware and Embedded Systems, CHES 2006, pp. 369- 383.

[5] K. Kursawe, A. Sadeghi, D. Schellekens, P. Tuyls, and B. Škorić, "Reconfigurable physical unclonable functions enabling technology for tamper-resistant storage", in 2nd IEEE Int. Workshop on Hardware-Oriented Security and Trust, HOST 2009, pp. 22-29.

[6] Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak. “Trusted integrated circuits: A nondestructive hidden characteristics extraction approach”, Lecture Notes in Computer Science, Springer-Berlin, vol. 5284, pp. 102-117, 2008.

[7] G.E. Suh and S. Devadas,”Physical unclonable functions for device authentication and secret key generation", in Proc. Design Automation Conference, DAC 2007.

[8] E. Ozturk, G. Hammouri, B. Sunar, “Physical unclonable function with tristate buffers”, in Proc. Int. Symp. on Circuits and Systems, ISCAS 2008, pp. 3194-3197, Washington, DC, USA.

[9] J. Guajardo , S. S. Kumar , G.-J. Schrijen , P. Tuyls, “FPGA Intrinsic PUFs and their Use for IP Protection”, in Proc. of the 9th Int. Workshop on Cryptographic Hardware and Embedded Systems, CHES 2007, Vienna, Austria.

[10] J. Guajardo, S.S. Kumar, G.-J. Schrijen, and P. Tuyls, “Physical unclonable functions, FPGAs and public-key crypto for IP protection,” in Proc. Int. Conf. on Field Programmable Logic and Applications, FPL 2007, August 27-29, 2007.

[11] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijenand, P. Tuyls, ”Extended abstract:The butterfly PUF protecting IP on every FPGA", in Proc. 1st IEEE Int. Workshop on Hardware-Oriented Security and Trust, HOST, 2008.

[12] Y. Su, J. Holleman, and B. Otis, “A digital 1.6 pJ/bit chip identification circuit using process variations,” IEEE Journal of Solid-State Circuits, vol. 43, no. 1, pp. 69–77, Jan. 2008.

[13] A. Maiti, and P. Schaumont, “Improving the quality of a physical unclonable function using configurable ring oscillators”, in Proc. Int. Conf. on Field Programmable Logic and Applications, FPL 2009, pp. 703-707, Washington, DC, USA.

[14] D. Lim.” Extracting secret keys from integrated circuits”. Master’s thesis, Massachusetts Institute of Technology, 2004.

[15] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas. “Controlled physical random functions”, in Proc. Annual Computer Security Applications Conference, ACSAC 2002, pp. 149–160, New York, 2002.

[16] M. Majzoobi, F. Koushanfar, and M. Potkonjak, “Techniques for design and implementation of secure reconfigurable PUFs,” ACM Trans. on Reconfigurable Technology and Systems, vol. 2, no. 1, pp. 1–33, 2009.

[17] C. W. O’Donnell, G. E. Suh and S. Devadas. “PUF-based Random Number Generation “, MIT CSAIL Technical Memo 481.

[18] B. Sunar, W. J. Martin, and D. R. Stinson, “A provably secure true random number generator with built-in tolerance to active attacks,” IEEE Trans. Comput., vol. 56, no. 1, pp. 109–119, 2007

[19] D. E. Holcomb, W. P. Burleson, and K. Fu, “Initial SRAM state as a fingerprint and source of true random numbers for RFID tags”, in Proc. Conf. on RFID Security, RFIDSec 07, July 2007.

[20] A.J. Acosta, M.J. Bellido, M. Valencia, A. Barriga, and J.L. Huertas, “Fully digital redundant random number generator in CMOS technology”, in Proc. of ESSCIRC 1993, pp. 198-201.

[21] Y. Dodis, M.Reyxin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data “, in Proc. of Eurcrypt 2004, vol. 3027 of Lecture Notes in Computer Science, Springer-Verlag, pp. 523-540.

[22] B. Gassend, “Physical random functions”, Masther´s thesis, Computer Science and Artificial Intelligence Laboratory, MIT, 2003.

[23] P. Tuyls, G.-J. Schrijen, F. Willems, T. Ignatenko, B. Skoric, "Secure key storage with PUFs," in Security with noisy data, T. Kevenaar, P. Tuyls, and B. Škorić (eds.), Springer, pp. 269-292, 2007.

[24] J. Bringerr, H. Chabanne, T. Icart: “On physical obfuscation of cryptographic algorithms", in vol. 5922 of Lecture Notes in Computer Science, Springer-Verlag , pp. 88-103, 2009.

[25] P. Tuyls, J. Guajardo, L. Batina, and T. Kerins, "Anti-Counterfeiting," in Security with noisy data, T. Kevenaar, P. Tuyls, and B. Škorić (eds.), Springer, pp. 293-312, 2007.

[26] G. Hammouri and B. Sunar, "PUF-HB: A tamper-resilient HB based authentication protocol", in Proc. of ACNS 2008, vol. 5037 of Lecture Notes in Computer Science, Springer-Verlag , pp. 346-365.

[27] N. J. Hopper and M. Blum,, “Secure human identification protocols" , in Advances in Cryptology, ASIACRYPT 2001, vol. 2248 of Lecture Notes in Computer Science, Springer-Verlag , pp. 52-66.

[28] S. Piramuthu, "HB and related lightweight authentication protocols for Secure RFID”, in Proc. CollECTeR Europe Conference, Basel, Switzerland, June 2006.

[29] L. S. Kulseng,“Lightweight mutual authentication, owner transfer, and secure protocols for RFID systems”, PhD Thesis, Iowa State University, 2009.