New Directions in Applying Physical Unclonable Functions

Rainer Falk and Steffen Fries
Corporate Technology
Siemens AG
Munich, Germany
e-mail: {rainer.falk|steffen.fries}@siemens.com

***Abstract*—Physical Unclonable Functions (PUF) realize the functionality of a "fingerprint" of a digital circuit. They can be used to authenticate devices without requiring a cryptographic authentication algorithm, or to determine a unique cryptographic key based on hardware-intrinsic, device-specific properties. It is also known to design PUF-based cryptographic protocols. This paper presents several new applications of PUFs. They can be used to check the integrity, or authenticity of presented data. A PUF can be used to build a digital tamper sensor. An identifying information in a communication protocol can be determined using a PUF, or a licensing mechanism can be realized.**

Keywords–physical unclonable function; key extraction; embedded security; licensing; configuration integrity

I. INTRODUCTION

The need for technical information technology (IT) security measures increases rapidly to protect products and solutions from manipulation and reverse engineering. Cryptographic IT security mechanisms have been known for many years, and are applied in smart devices (internet of things, industrial and energy automation). Such mechanisms target authentication, system and communication integrity and confidentiality of data in transit or at rest. One base for the operation of security mechanisms is typically a cryptographic key that has to be stored securely on devices. Upcoming industrial security standards (ISO/IEC 62443 [1]) require explicitly a hardware-bound storage for cryptographic keys.

A significant effort is often required in practical realizations to protect key storage, e.g., by external hardware integrated circuits (IC). In current research, IT security methods are investigated that directly use a unique physical property of an object as a physical fingerprint. Small random differences of physical properties are used to identify an object directly, or to derive a cryptographic key for conventional cryptographic IT security mechanisms [2]. A digital circuit, i.e., a semiconductor integrated circuit, can contain a digital circuit element called a physical unclonable function (PUF) to determine the physical device fingerprint. Minimal differences in the semiconductor structure, like for instance the doping of a semiconductor, the layer thickness, or the width of lines arise at the production randomly. This is similar to the random surface structure of paper sheets. These chip individual properties are "simply there" without being designed-in explicitly, or being programmed by a manufacturer during production. Such a device fingerprint is unique, and cannot practically be reproduced easily (unclonability). In addition, the fingerprint can be modified, or even destroyed when the IC is manipulated.

After giving an overview of some major realization possibilities for a digital PUF in Section II, basic usages of a PUF are summarized in section III. The main contribution of the paper is in section IV, describing several new applications of PUF technology. Section V concludes with a summary, and an outlook.

II. PHYSICAL UNCLONABLE FUNCTIONS AS DIGITAL DEVICE FINGERPRINT

A PUF can be realized on a semiconductor circuit to determine a device-specific piece of information depending on variations in the target physics due to the manufacturing process. The information provided by the PUF can be used directly for low-cost authentication, to determine a serial number as an identifier, or as cryptographic key. The semiconductor circuit can be an application-specific integrated circuit (ASIC), or a field-programmable gate array (FPGA). This section gives a short overview about PUFs. More detailed information is available in tutorials on PUFs [3][4][5][6].

PUFs have been a major topic of academic research. However, PUF technology is already applied commercially. Examples are Intrinsic ID [7], Verayo [8][9], Microsemi Smartfusion2 FPGAs [10], and NXP SmartMX2 [11].

Common digital circuits are designed to provide identical behavior on different ICs. However, a PUF circuit is designed to provide different results on different ICs, but identical or at least similar results on the same IC when the function is executed repeatedly.

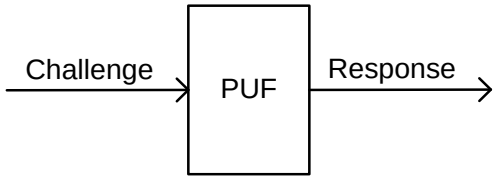

Figure 1. Challenge-Response-PUF

Figure 1 shows a challenge-response PUF, in which the PUF circuit determines a response value depending on a provided challenge value. Weak PUFs and strong PUFs are distinguished: while a strong PUF has a wide range of challenge input values, a weak PUF has no, or only a very

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5

limited set of challenge values. A strong digital PUF can be realized by reconfiguring a digital PUF circuit depending on the challenge value.

The objective of a PUF circuit is that on the same IC, the response value for a given challenge value is stable (reproducibility), while on different ICs, the response values are different (uniqueness). As binary values are used for challenge and response values, the similarity can be measured by the Hamming weight, i.e., the number of different bits. The measure for reproducibility is the intra-device Hamming distance, i.e., the mean value of the number of different bits when the PUF is executed multiple times for a given challenge value. The measure for the uniqueness is the inter-device Hamming distance, i.e., the mean value of the number of different bits when executed in different ICs.

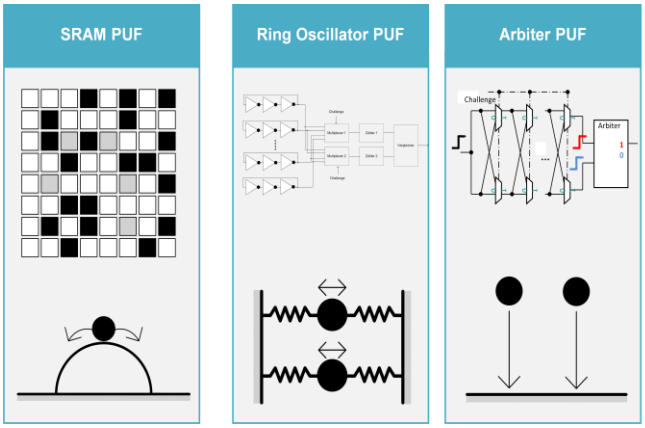

Figure 2. Example PUF Realizations and Their Mechanical Analogon

Figure 2 shows three examples of well-known constructions of PUFs and their mechanical analogon:

- SRAM-PUF: power-up value of static random-access memory (SRAM) cells
- RO-PUF (Ring oscillator PUF): oscillator frequency
- Arbiter PUF: time delay

Many more constructions for a digital PUF have been proposed, e.g., bi-stable Ring PUF, Flip-Flop-PUF, Glitch PUF, Cellular Non-linear Network PUF, or Butterfly-PUF.

A. *SRAM-PUF*

A digital memory can store binary values 0 and 1. After power-up, some memories show a device-specific initialization pattern. The power-up value of a memory cell can be either 0 or 1, or being instable (sometimes 0, sometimes 1). The pattern of power-up values of its memory cells is characteristic of a memory IC, depending on small variations of the semiconductor physics of each memory cell.

A mechanical analogon for the power-up is a ball placed on the top of a hill [12]. When the whole geometry is exactly symmetric, the ball will roll-down to the left side and to the right side with the same probability. If the hill, or the ball, would have some asymmetries from manufacturing, the ball will tend to roll-down either to the left side or to the right side.

B. *Ring Oscillator PUF (RO-PUF)*

A digital circuit can realize an oscillator using a delay circuit with a feedback loop (ring oscillator). The oscillation frequency depends on manufacturing variations. The frequency of two identically designed oscillators can be compared using a counter, and comparator. Depending on the IC, one or the other will oscillate with a higher frequency. Realizing multiple oscillators, a “fingerprint” of the digital circuit can be obtained.

A mechanical analogon is an oscillating mass, and spring. Two identical physical realizations will in practice have a slightly different oscillation frequency, depending on small physical variations.

C. *Arbiter PUF*

A further effect that can be used to build a PUF is time delay. Two identically designed signal paths will show minimal differences in the respective delay. After giving in input signal to both signal paths at the same time, an arbiter circuit determines the faster signal path, i.e., the signal path on which the signal appears first,

A mechanical analogon is a drop test for two identically manufactured masses. Depending on variations in the height, or the surface of the masses, one will tend to impact first on the floor.

III. BASIC PUF APPLICATIONS

A PUF can be used for security purposes in different ways. It can be used as low-cost object authentication, or to determine a cryptographic key. This section describes these two basic applications, and gives examples for some specific usages of PUFs.

A. *Object Authentication*

Authentication is an elementary security service proving that an entity in fact possesses a claimed identity. Often natural persons are authenticated. The basic approaches a person can use to prove a claimed identity are by something the person knows (e.g., a password), by showing something the person has (e.g., passport, authentication token, smart card), or by exposing a physical property the person has (biometric property, e.g., a fingerprint, voice, iris, or behavior). Considering the threat of counterfeited products (e.g., consumables, replacement parts) and the increasing importance of ubiquitous machine-based communication, also physical objects need to be authenticated in a secure way. Various different technologies are used to verify the authenticity of products, e.g., applying visible and hidden markers, using security labels (using, e.g., security ink or holograms), and by integrating cryptographic authentication functionality in wired product authentication tokens, or Radio Frequency Identification (RFID) authentication tags.

An object or digital circuit can be identified by a serial number. For authentication, a cryptographic authentication protocol can be used, requiring a secret/private key to be available on the object to be authenticated.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5

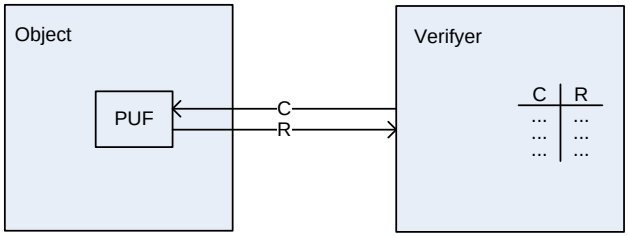

Figure 3. Challenge-Response-Authentisierung

For authentication, a challenge value is sent to the object to be authenticated. A corresponding response value is sent back and verified. The response is determined by the PUF. As only an original product can determine the correct response value corresponding to a challenge, the product entity or a dedicated part of the product is thereby authenticated.

Figure 3 shows how an object becomes authenticated by a verifier. The verifier maintains a database of reference challenge response pairs. For example, the database was filled during production of the object by recording arbitrary challenge-response-pairs. During the authentication the verifier selects a challenge value of the database and sends it to the object to be authenticated. The response value R is determined by means of the PUF, and transferred back to the verifier. The verifier compares the received response value with the reference value stored in the database. If these are similar, i.e., the number of different bits does not exceed a threshold, the object as authenticated successfully.

B. Cryptrographic Key Extaction

A cryptographic key can be determined based on inexact, noisy data. A "fuzzy key extractor" is a functionality that determines a stable cryptographic key using a PUF, and helper data [13][14]. The helper data allows to correct bit errors of responses (noisy data), and to map the PUF output to a given cryptographic key. A main advantage is that no secure non-volatile memory is needed on the device to store a cryptographic key.

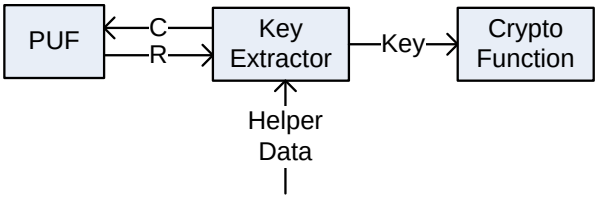

Figure 4. PUF Key Extraction

The PUF is used internally within a digital circuit to determine response values, see Figure 4. The helper data does not have to be stored securely. It can be used only by a single IC to determine the cryptographic key on the device.

C. Further PUF Applications

Several further applications, besides the two basic PUF usages stated above, have been proposed and designed. The following list section gives an overview on related work.

- A PUF can be used to prevent utilizing specific features of semiconductor ICs. Without chip-specific aiding information, the performance of an IC is reduced or access to certain memory partitions is prevented. Also, a PUF can be used to bind software intellectual property to a FPGA device by encrypting the software code using a PUF-generated device key [15], which is typically done during manufacturing. This solution can be used to protect for instance remote software updates [16].
- A PUF can be used to protect the execution of software code: the Control Flow Graph of an executed program depends on the output of a PUF [17].
- It is known to include a measurement value determined by a sensor as part of the challenge of a PUF to authenticate the sensor measurement [18][19]. This allows authenticating sensor measurements.
- A PUF can be used also in data communication to determine a message integrity checksum (message integrity code, message authentication code) [20]. While a real, physical PUF is used to determine the message authentication code by the sender, a simulated, algorithmic model of the PUF is used to verify the checksum by the receiver.
- Furthermore, the cryptographic key derived by a PUF of a semiconductor can be used to decrypt configuration data [21].
- A PUF can, as security primitive, be integrated in a cryptographic protocol directly [22][23].

D. Limitations of PUF

Building security solutions using PUFs, it is important to understand their limitations. Important issues to be considered are:

- Attacks on PUFs, and support functions as a fuzzy key extractor need to be taken into account. This relates for instance to the PUF model building, potential side channel attacks, and also fault injection attacks
- Robustness of a PUF implementations with respect to tamper resistance, e.g., how vulnerable is a solution with opened chip housing
- Reliability of the PUF with respect to the long term application in devices related to ageing, environmental conditions as temperature and others.
- Required processes for enrollment of data, which relates on one hand to the helper data on device, and within backend systems. On the other hand, depending on the PUF application the handling of the recorded challenge response pairs needs to be defined, as this information is sensitive and can be system critical. The latter may be compared to the handling of symmetric device keys, which have a similar level of sensitivity.

Based on these points it becomes even more obvious, that a security solution exposing the PUF functionality to other elements needs to designed PUF aware, especially

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5

considering reliability and resilience requirements for long lasting deployments.

IV. New Applications of Phyiscal Unclonable Functions

Applications of PUFs fall basically in two categories: challenge response authentication, e.g., for low cost RFID Tags, and extraction of a symmetric cryptographic key. When used for protecting embedded systems, the cryptographic key can be used independent of PUF properties.

In this section, we describe potential new applications of PUFs in the context of security services.

A. Authentication Verification

It is known to use a PUF to authenticate an integrated circuit or a device respectively. However, the reverse is possible as well: the PUF can be used to verify the authentication of an external party. This approach has the clear advantage that no cryptographic algorithm has to be implemented to perform authentication checks. It rather requires the storage of a certain number of challenge-response pairs.

One application for this authentication verification can be, e.g., in the context access verification to a diagnosis or debug interface of an integrated circuit, or to protect the wake-up functionality offered by these chips.

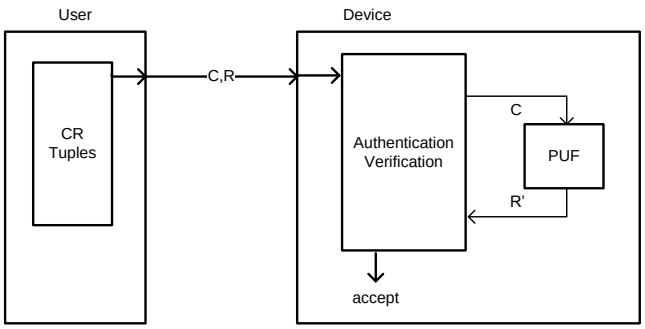

Figure 5. PUF Authentication Verification

Figure 5 shows how a PUF can be used to check authenticated access to a device or device functionality: a user presents a C/R pair of challenge C and response R. The PUF determines the response R' for the given challenge C. If the presented response R, and the determined response R' are identical or differ only in a limited number of bits, access is granted (accept). The C/R pair can be determined in different ways:

- In an initialization phase, C/R pairs can be read out from the device, and stored in a secure data base. Before the IC is put in operation, the interface to read out C/R pairs is blocked, e.g., by burning a security fuse.
- Should the PUF be a PUF for which an algorithmic model can be determined (as described in [20], the algorithmic model of the PUF can be used to compute C/R pairs.

B. Configuration Integrity Check

In a similar way, the integrity of configuration data can be verified by a PUF, directly.

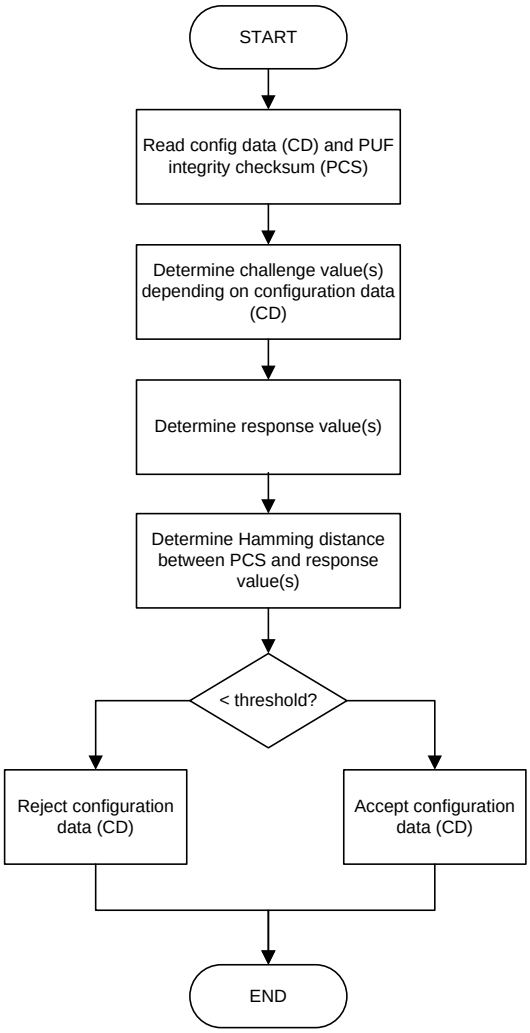

Figure 6. PUF-based Integrity Check of Configuration Data

Figure 6 shows a realization option: configuration data is read from an external, unprotected configuration memory, e.g., a serial electrically erasable programmable read only memory (EEPROM). Besides the configuration data (CD), a PUF checksum (PCS) is also read. A PUF challenge value is determined depending on the configuration data, e.g., a cryptographic hash value of the configuration data. The corresponding response value R' is determined, and the Hamming distance between R' and the PCS value is determined. The configuration data are accepted if the number of different bits is below a given threshold value.

A PUF may be used in a similar way as a key derivation function for a cryptographic key *K*. Depending on the cryptographic key *K*, challenge values are determined. The PUF response value(s) are used to determine a (derived) key.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5

C. PUF Tamper Sensor and PUF Built-In Self Test

Challenge response pairs of the PUF are typically stored as reference data. The integrated circuit uses the reference data to check whether the PUF is working correctly. This can be used for different purposes:

- A PUF-based tamper-sensor can be realized: when a tampering of the device occurred, the PUF provides different response value with high probability.
- A built-in self test functionality can be realized for a PUF, used, e.g., for authentication, or key extraction. Only if the PUF works as expected, the self-test succeeds.

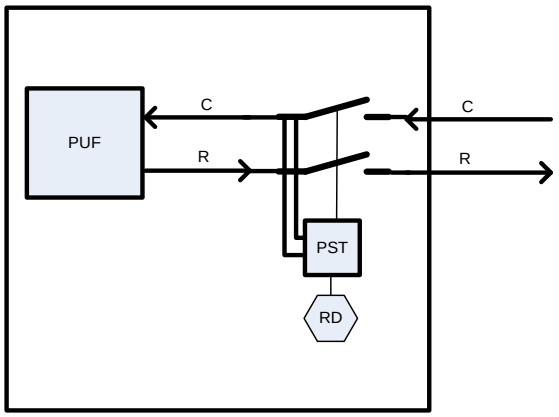

Figure 7. PUF Built-In Self Test

Figure 7 shows a realization option where reference data (RD) are used to check the PUF. Only if the PUF provides responses sufficiently similar to the reference data, access to the PUF is enabled by the PUF self-test unit PST.

D. Identifying Communication Sender

A PUF can be used to derive a serial number of a device. This PUF derived serial number or a derivation of thereof can be used to determine an identifier for data communication.

For example, an IPv6 stateless address auto configuration can be performed using a PUF. T. Aura defines how an IPv6 address can be created cryptographically [24]. Similarly, a PUF can be used to determine an IPv6 address. The challenge can be determined based on network part of the IPv6 address assigned by an IPv6 router. The host part is created depending on the PUF response output.

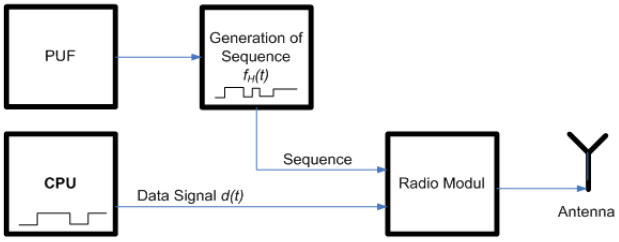

Figure 8. PUF-based Spread-spectrum Transmission

Figure 8 shows a different variant where the PUF-based identifying information is not included in the sender address. Instead, if a wireless spread spectrum transmission system is used, a spreading code is build or modified respectively depending on the PUF response. Hence, the PUF is used to realize a kind of "stream cipher" as spreading code.

E. PUF-helper Data as License File (license key)

A fuzzy key extractor allows determining a given cryptographic key using helper data. The helper data has two purposes: it allows correcting random errors of the PUF response, and it transforms the device-specific PUF response to a given cryptographic key. These properties can be used as licensing mechanism.

In a licensing scheme, a license code, or license key, is required to use a certain, software-based feature. The license code/key can be checked, resp. a key to decrypt code can be determined based on the license key.

With PUF helper data, the license code/key can be provided in the form of helper data: as long as the required helper data is not available, the license key cannot be built by a device. However, if helper data to reconstruct a certain license code/key is provided, the device can determine the license code/key. As a PUF is used, the helper data can be processed only on the single intended target device.

V. CONCLUSION

Physical unclonable functions have been investigated extensively by both research, and industry. The work focuses much on design constructions to realize a PUF, analyzing their statistical, and security properties, and on key extraction. Although being known for at least 10 years, one limited number of examples for commercial applications exists. Besides the classical usages, object authentication, and key extraction, a PUF can be specific new usages can be realized based on a PUF. This paper described several new applications for PUFs in different systems, either self-contained, like the tamper sensor or in conjunction with other parts of target solutions like in the case of licensing. These new applications are discussed as abstract concepts and need to be investigated and realized to gain more experience about the actual feasibility in products or solutions. This work is envisioned for the future.

Issues for the practical application are the stability over time (ageing), and under changing environmental conditions. As PUFs are still a relatively new security feature that is not yet broadly applied in practice, careful analysis of the actual security level as to be performed (e.g., modeling attacks, physical attacks, side channel attacks). The security management of PUF-based security solution has to be designed (e.g., enrollment of key material, building and maintaining databases comprising challenge/response pairs).

However, PUFs show unique properties that make them interesting for practical usage: they allow "storing" a cryptographic key in a protected way without requiring physical non-volatile memory. Low-cost authentication solutions can be built that do not require implementations of cryptographic algorithms.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5

REFERENCES

[1] ISO/IEC 62443, "Industrial Automation and Control System Security" (formerly ISA99), available from: http://isa99.isa.org/Documents/Forms/AllItems.aspx, last access: January 2015

[2] B. Gassend, "Physical Random Functions", Masters Thesis, MIT, February, 2003, available from: http://csg.csail.mit.edu/pubs/memos/Memo-458/memo-458.pdf, last access: January 2015

[3] C. Herder, Y. Meng-Day, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial", Proceedings of the IEEE, Vol.: 102 No. 8, Aug. 2014, pp. 1126-1141, available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6823677, last access: January 2015

[4] S. Devadas, "Physical Unclonable Functions and Applications, Presentation Slides", available from: http://people.csail.mit.edu/rudolph/Teaching/Lectures/Security/Lecture-Security-PUFs-2.pdf, last access: January 2015

[5] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation", DAC 2007, June 4-8, 2007, pp. 9-14 ACM, available from: http://www.verayo.com/pdf/2007_PUF_dac.pdf, last access: January 2015

[6] S. Katzenbeisser, Ü. Kocabas, V. Rožic, A. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon", IACR eprint 2012/557, Sep. 2012. Online]. Available from: https://eprint.iacr.org/2012/557.pdf, last access: January 2015

[7] Intrinsic ID Technology, available from: https://www.intrinsic-id.com/technology/, last access: January 2015

[8] Verayo, "Physical Unclonabel Functions (PUF)", available from: http://verayo.com/tech.php, last access: January 2015

[9] Verayo, "Introduction to Verayo", available from: http://www.rfidsecurityalliance.org/docs/Verayo_Introduction_RFIDSA_July_9_08.pdf , last access: January 2015

[10] Microsemi, "SmartFusion2", available from: http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2, last access: January 2015

[11] NXP, "NXP Strengthens SmartMX2 Security Chips with PUF Anti-Cloning Technology", February 2013, available from: http://www.nxp.com/news/press-releases/2013/02/nxp-strengthens-smartmx2-security-chips-with-puf-anti-cloning-technology.html, last access: January 2015

[12] C. Böhm and M. Hofer, "Physical Unclonable Functions in Theory and Practice", Springer, 2012

[13] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data", Eurocrypt 2004, LNCS 3027, Springer, 2004, pp. 523-540, available from: http://www.iacr.org/archive/eurocrypt2004/30270518/DRS-ec2004-final.pdf, last access: January 2015

[14] B. Škorić, P. Tuyls, and W. Ophey, "Robust key extraction from Physical Uncloneable Functions", Applied Cryptography and Network Security, LNCS 3531, Springer, 2005, pp 407-422, available from: http://members.home.nl/skoric/security/PUF_KeyExtraction.pdf) , last access: January 2015

[15] Y. Alkabani and F. Koushanfar, "Active Hardware Metering for Intellectual Property Protection and Security", 16th USENIX Security Symposium, 2007, pp. 20:1-20:16, available from: http://www.usenix.org/event/sec07/tech/full_papers/alkabani/alkabani.pdf, last access: January 2015

[16] M. Gora, A. Maiti, and P. Schaumont, "A Flexible Design Flow for Software IP Binding in FPGA", IEEE Transactions on Industrial Informatics, vol. 6, issue 4, Nov. 2010, pp. 719-728

[17] R. Nithyanand and J. Solis, "Theoretical Analysis: Physical Unclonable Functions and the Software Protection Problem", IEEE Symposium on Security and Privacy Workshop, 2012, pp. 1-11 available from: http://www.ieee-security.org/TC/SPW2012/proceedings/4740a001.pdf, last access: February 2015

[18] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling Attacks on Physical Unclonable Functions", Proc. of the 17th ACM conference on Computer and communications security, 2010, pp. 237-249, , available from: http://people.idsia.ch/~juergen/attack2010puf.pdf, last access: January 2015

[19] K. Rosenfeld, E. Gavas, and R. Karri, "Sensor Physical Unclonable Functions", IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), June 2010, pp. 112-117, available from: http://isis.poly.edu/~kurt/papers/sensorpuf.pdf, last access: January 2015

[20] W. Bares, S. Devadas, V. Khandelwal, Z. Paral, R. Sowell, and T. Zhou, "Soft message signing", patent application, WO2012154409, Nov. 2012

[21] S. Devadas and T. Ziola, "Securely field configurable device", patent application, US2010/0272255, Oct. 2010

[22] M. van Dijk and U. Rührmair, "Physical Unclonable Functions in Cryptographic Protocols: Security Proofs and Impossibility Results", Cryptology ePrint Archive: Report 2012/228, April 2012, available from: https://eprint.iacr.org/2012/228.pdf, last access: January 2015

[23] M. Majzoobi, M. Rostami, F. Koushanfar, D. Wallach, and S. Devadas, "Slender PUF Protocol: A lightweight, robust, and secure authentication by substring matching", IEEE CS Security and Privacy Workshop, 2012, pp. 33-44, available from: http://www.ieee-security.org/TC/SPW2012/proceedings/4740a033.pdf, last access: March 2015

[24] T. Aura, "Cryptographically Generated Addresses (CGA)", RFC3972, March 2005, available from: https://www.ietf.org/rfc/rfc3972.txt, last access: January 2015

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5