A new Definition and Classification of Physical Unclonable Functions

Rainer Plaga
Federal Office for Information Security (BSI)
Godesberger Allee 185-189
Bonn, Germany
rainer.plaga@bsi.bund.de

Dominik Merli
Fraunhofer Research Institution AISEC
Parkring 4
Garching (near Munich), Germany
dominik.merli@aisec.fraunhofer.de

ABSTRACT

A new definition of "Physical Unclonable Functions" (PUFs), the first one that fully captures its intuitive idea among experts, is presented. A PUF is an information-storage system with a security mechanism that is
1. meant to impede the duplication of a precisely described storage-functionality in another, separate system and
2. remains effective against an attacker with temporary access to the whole original system.
A novel classification scheme of the security objectives and mechanisms of PUFs is proposed and its usefulness to aid future research and security evaluation is demonstrated. One class of PUF security mechanisms that prevents an attacker to apply all addresses at which secrets are stored in the information-storage system, is shown to be closely analogous to cryptographic encryption. Its development marks the dawn of a new fundamental primitive of hardware-security engineering: cryptostorage. These results firmly establish PUFs as a fundamental concept of hardware security.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Physical Security; H.3.0 [**Information Storage and Retrieval**]: General

General Terms

Security

Keywords

ACM proceedings, Physical Unclonable Functions

1. INTRODUCTION

Physical Unclonable Functions (PUFs)[1] were originally understood as "hardware devices that are hard to characterize and can be uniquely identified" [1]. They have developed into an important research topic in the field of hardware security within the past decade. There have been a number

of attempts (summarized in section Section 2.1) to precisely define what a member of the research community would intuitively call a PUF. Searched for is a definition that contains a *complete and minimal* list of the conditions that have to be fulfilled to identify a device as a PUF. No necessary condition must be absent. No superfluous condition, i.e. one that PUFs can, but do not absolutely have to fulfill, must be present. In section 2.2 we present a novel definition of a PUF. We demonstrate that it really fulfills, for the first time, both of the above demands.

A novel classification scheme for the PUF security objectives and mechanisms is presented in Section 3. Its usefulness is demonstrated by two examples. In Section 4 we show that the PUF security mechanism "cryptostorage" has a fundamental importance similar to the one of cryptography. PUF research is thus of a much more fundamental importance than hitherto realized. We discuss how the new scheme motivates qualitatively novel questions for further research in Section 5 and helps the certification of PUFs in Section 6 respectively. Section 7 concludes this paper.

2. DEFINITION OF PUFS

2.1 Previous Proposals

In the following, we briefly sketch the development of the ideas on which we base our proposal. Starting with an early paper by Gassend et al. [1], most proposed definitions are based on the concept of a physical challenge-response function which can be evaluated in a reproducible manner. Until approx. 2010, most proposed definitions characterized PUFs further by being "hard to characterize" [1], "hard to predict" [2], "physically unclonable" [3] or "tamper resistant" [4]. Such demands carry the problem that devices which turn out to be predictable, clonable or tamper vulnerable are no longer PUFs according to the proposed definitions. However, e.g., in a certification process, the concept of an "insecure, broken PUF" is clearly necessary. Indeed, in practice, the community does not really employ such definitions because it universally continues to refer to insecure PUFs that have been completely broken as "PUFs". Armknecht et al. [5] recognized this problem and defined PUFs as physical functions PFs, i.e., they dropped the U for "unclonable". Such physical functions are digital memories because the latter are physical devices that always have to realize a challenge-response mechanism: an address is applied as a challenge and the memory content is returned as a response. The remaining problem is then: how to characterize the *specific characteristic of PUFs* that sets it apart,

e.g., from a standard memory stick, which is, of course, not understood as a PUF in the community? An alternative to identify it with absolute unclonability is to identify it with certain features of the PUF architecture. Several definitions [6, 7, 8] propose to identify the specific characteristic properties that we call below in Section 3 "complex-structure upon production" and "cryptostorage". We will discuss in Section 3.1.2 that these properties classify special security mechanisms. But a demand for a special security mechanism in its definition would render the PUF concept inflexible. E.g. the information stored in a quantum token, discussed in section 3.2, must be loaded after its production - otherwise the storage is not secure. A definition that demands a loading upon production would rather arbitrarily excludes quantum tokens as PUFs.

We will explicitly identify the *specific characteristic* of PFs, that we searched for in the previous paragraph in section 2.2.2.

2.2 Proposed Definition of PUFs and its Meaning

2.2.1 Auxiliary and PUF Definition

We first formulate an auxiliary definition:

- **Definition of a physical information-storage system (1)**
  An information-storage system is a set of at least two modules[1] that are conceptually or physically permanently connected. These modules (or parts of the module if the system is a single module) have the following purposes: a "storage module" can be put into a physical state H which is determined by the information in a challenge C. This storage module is measured in this state and the measurement result is reproducibly encoded as the information of a response R by another "encoder module" of the total system. The set of all Rs for all Cs is the information stored in the system.

This definition agrees with the intuitive idea of a digital memory, and makes its relation to a challenge-response function precise. The challenges are the addresses at which the physical information, the responses, are stored. An information-storage system is depicted in Figure 1. The term "reproducibly encoded" demands that a noise in the responses is bounded.
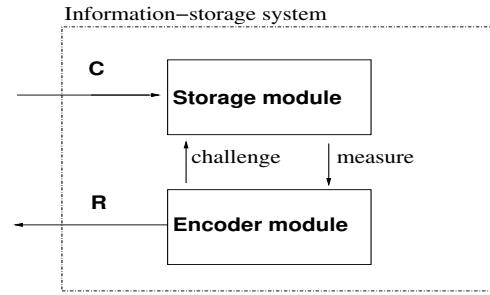
- **Definition of a PUF (2)**
  A PUF is a physical information-storage system that is protected by a security mechanism which

  1. *has the security objective to render it more difficult to duplicate a precisely described storage functionality of the system in another, separate system.*
  2. *is meant to remain effective against an active[2] attacker with temporary physical access to the whole*

---

[1] A module is defined to be an artificial (the artificiality is necessary to delimit PUFs from biometric systems, a closely related concept with which the PUF concept is often compared) physical system that is permanently physically connected and serves a certain purpose. A physical system is defined to be a set of materials and fields that is delimited from the rest of the world in a well defined manner.

[2] Active means that she can change the system rather than only passively listen.

Information–storage system

Figure 1: Sketch of a physical information-storage system, according to our definition. The dot-dashed outer box symbolizes the whole system which usually is (but does not have to be) also a module, like the storage- and encode-module always are.

system in its original form.

2.2.2 Explanation of the PUF Definition

The definition does not demand that a security-mechanism actually does render the duplication more difficult, but only that an implemented mechanism is meant to do so. This avoids the circularity problem with earlier PUF definitions discussed in Section 1 ("an insecure PUF is no PUF"). As the following example shows security primitives are often defined in an analogous manner:

- **Definition of cryptographic encryption algorithm (3)**
  A cryptographic encryption algorithm is a key-dependent mapping from a cleartext string of bits to a cryptogram string of bits. Its security objective is that an attacker finds no algorithm for an inverse mapping (decryption) of the cryptogram to the cleartext without the key. The key is a string of bits on which the encryption algorithm depends.

Caesar's cipher is trivial to break, but still a cryptographic encryption algorithm because there is a specific key dependent mapping with the said objective. The precise storage functionality is the most characteristic feature of a PUF. It can go beyond the mere storage and release of information, as in the case of public PUFs [9, 10], where a release of the response within a certain time frame is required. Even if the functionality is merely the storage and release of physical information, the objective can be to prevent duplication to different degrees as in the previous proposals of a "physical" and "mathematical" duplication [3]. We will come back to the classification of PUF security objectives in Section 3.1.1. We define an "information-storage system" as the system that actually stores the information (rather than auxiliary systems for packaging, energy supply etc.). Our definition of a PUF requires that the security mechanism is based on the system's storage-mechanism, because otherwise it would not remain effective against an attacker with full access to the original system. This makes the required security- and information-storage mechanism indivisible, i.e., without the latter, the former cannot be realized. Devices in which information-storage and security mechanism are separable are no PUFs. This indivisibility, let us call it "security-memory boundedness", is the *specific characteristic* of a PUF that was searched for in Section 2.1.

Another angle on the motivation for the second condition of the PUF definition will be discussed in Section 2.2.4.

A main result of this paper is that the PUF definition list is complete and minimal with the proposed conditions. No other of the myriad other conditions that PUFs may well fulfill, e.g., that the storage mechanism is noisy, that the security mechanism works without energy supply, etc., has to be fulfilled to characterize a module as a PUF. Summarizing we state:

A module is a PUF if and only if it fulfills both conditions of the definition (2).
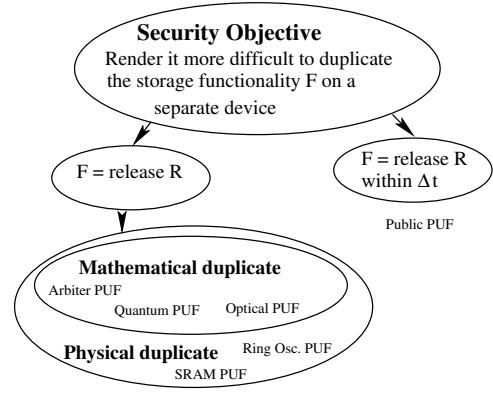
2.2.3 Exemplary Analysis: Arbiter PUF Characterized as a PUF

As an exemplary case, we discuss how the arbiter PUF [11] is characterized as a PUF in the sense of our definition. The arbiter PUF has a storage module in the form of a chain of multiplexers that are programmed by a certain number of inputs which act as a challenge. The encoder module is a latch that determines which of two delay paths - that are configured (with other words "put in a physical state") by the challenge - produced a longer delay for two test signals. The arbiter PUF is thus an information-storage system in our sense. The security mechanism is described by Suh and Devadas as: "a set of challenges (is mapped) to a set of responses based on an intractably complex physical system." The unclonable storage-functionality is the release of a response after a challenge is supplied. The authors make clear that this must be understood as an objective rather than a property because "to prevent model-building attacks" a further development of the PUF architecture might be needed. Thus the first condition of our definition of a PUF is fulfilled. The "intractably complex structure" upon production is meant to protect against an attacker with temporary access to the storage system. Therefore, the second condition of our definition (2) is fulfilled, too.

2.2.4 Delimitation of PUFs from other Concepts

Conventional secure memories, as typically realized, e.g., in smartcards as conventional digital memories with some passive or active protection shields to guarantee "tamper resistance", are no PUFs because they are not security-memory bounded. A conventionally secured memory has been called "controlled" by Gassend et al.[12] [3]. The qualitative novelty of PUFs, characterized by the second condition in the definition (2), is that their security mechanisms go beyond such conventional (access) control.

Unique physical labels as they are used, e.g., on banknotes or drug packages against counterfeiting are no PUFs, not conceptually permanently connected to an encoder module as required by our definition (2). "Physically obfuscated keys" [13] with a read-out mechanism for the key are PUFs because their obfuscation through a non-standard storage mechanism is security-memory bound.

---

[3]Gassend et al. call a PUF controlled "if it can only be accessed via an algorithm that is linked to the PUF in an inseparable way (i.e. any attempt to circumvent the algorithm will lead to the destruction of the PUF." If we replace "PUF" by "information-storage system" and interpret "algorithm" as "hardware-interface that implements the system's intended functionality", this is a definition of the hitherto conventional manner to protect an information-storage system against duplication.

Figure 2: Classification of the PUFs' security objectives; some existing PUFs are assigned to these classes.

3. PUF CLASSIFICATION

3.1 Classification Criteria

3.1.1 Security Objectives

The most basic classification is by security objectives. According to the first condition in definition (2) the objectives are characterized by the precise nature of the storage functionalities and the kind of duplication to be prevented. The simplest storage functionalities are:

S1. To store physical information and to release it upon some challenge which can be:

> *S1.a. a simple trigger or*

> *S1.b. a sophisticated address, chosen by the user.*

An example for a more sophisticated storage functionality is:

S2. the release of the stored information within a certain time interval Δt.

There are two widely used meanings of the term "duplicate"[3].

D1. "Physical duplication" is to prevent a duplication in physical detail. I.e. the separate duplicated system either follows the specifications for the construction of the original, or at least returns physically identical responses.

D2. "Mathematical duplication" only requires that the duplication system returns responses with the same information content as the ones of the original module to all challenges. Its physical structure is otherwise not constrained.

D2 is a more ambitious objective than D1 because it requires to absolutely prevent the readout of the stored information at the challenge addresses, also against sophisticated modeling attacks[18]. As soon as this information is read out, it can be stored in another memory under the same address thus realizing a mathematical clone. D1 can still be attainable when the stored information is known to the attacker. Fig. 2 illustrates the storage-functionality classes and gives examples of PUFs within them.

The security objectives are determined by the security architecture that uses the PUF rather than the architecture of the PUF itself. In Section 3.2, we will discuss classification examples for some PUFs including one that can have both objective D1 or D2 in different contexts.

3.1.2 Security Mechanisms

Three principal classes of security mechanisms for PUFs have been proposed up to now.

Security mechanism 1 "Complex Structure"(CS). A complex structure of the information-storage module prevents analysis and/or reproduction.

Most PUFs that were proposed up to now rely on this principle. Usually the following principle is exploited: it is easier to create a complex random structure, than to analyze and duplicate it. Such PUFs are produced in a process that creates some complex structure in the storage module from which random responses can be derived by the encode module[14]. We classify this security mechanism as PUFs with "Complex-Structure upon Production" (CSP) property.

Security mechanism 2 "No Cloning"(NC). A principle of physics forbids the duplication of the storage module.

This is a security mechanism that prevents the duplication of the storage unit because of some fundamental principle of physics. An example are physical quantum systems in a state unknown to the attacker, that cannot be cloned due to the "no cloning principle"[15].

Security Mechanism 3 "Cryptostorage". Let us call a subset of all possible challenges the set of secret challenges (s-challenges). The secret responses (s-responses) to be protected are defined as the responses to the s-challenges. The security mechanism prevents that the attacker can apply all or most s-challenges.

If the attacker cannot apply an s-challenge in principle she will not be able to get the corresponding secret (response), i.e. cryptostorage protects the physical stored information against read out. Cryptostorage is closely analogous to cryptographic encryption. This parallel will be further developed in Section 4.

Two general architectures to realize cryptostorage in practice have been proposed [8].

Security mechanism 3a. "Minimum Readout Time (MRT)". The number of challenge-response pairs is chosen very large. Then, if there is a sufficiently long minimum readout time for one challenge-response pair, the attacker cannot apply the s-challenges within a reasonable time period if the set of s-challenges is unknown to her.
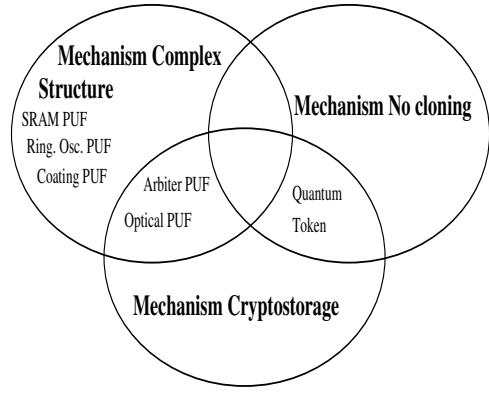
PUFs with a MRT-mechanism are a.k.a. as "strong" PUFs [13].

Security mechanism 3b. "Erasure Upon Readout (EUR)". If a non s-challenge is applied the s-response corresponding to the s-challenge is erased. Therefore the attacker cannot apply the s-challenge if it is not known to her, because an attempt to do so, erases most of the secret information in general.

Fig. 3 illustrates which PUF constructions are protected by these mechanisms.

3.2 Exemplary Classification of some Existing PUFs

The ring-oscillator PUF [11] and the SRAM PUF [16] exploit variations in ring oscillator frequencies or memory cell balances respectively, to generate secret bits and store the information in a currently unconventional manner. The security mechanism is to generate a random value during fabrication i.e. it is a CSP mechanism. The security objective of the ring-oscillator PUF [11] and the SRAM PUF [16] (a.k.a "weak PUFs" [13]) can be twofold. One possible aim is to prevent the physical duplication (security objective D1 in

Figure 3: The PUF security mechanisms and PUFs that are based on them. See Section 3.1.2 for further explanation.

Section 3.1.1). This security objective will typically arise in a context in which one aims to make it difficult for the producer of the PUF to produce physical duplicates. The other possible aim is to render the mathematical duplication (security objective D2 in Section 3.1.1) more difficult by making a readout in the switched-off state more difficult. This second aim cannot reasonably be to prevent the read-out of the stored information altogether because no security mechanism of SRAM and ring-oscillators prevents such readout. The arbiter PUF [17] uses intrinsic delay variations in microchips and was proposed to be used with an s-challenge that is much smaller than the set of all challenges. It thus aims to prevent the mathematical duplication (security objective D2, in Section 3.1.1) with the help of both the CSP mechanism and cryptostorage with the MRT mechanism. For quantum-token based PUFs [15], the security objective D2 is reached via proving that the s-challenge is necessary to extract the previously stored information. Therefore, this PUF prevents mathematical duplication with the help of the NC mechanism and cryptostorage with the EUR mechanism.

A qualitatively special authentication functionality is realized with the "public PUF" [9, 10] idea, which proposes PUFs that can prove their authenticity even though their challenge-response pairs are public. A public PUF not only stores and protects information, but is also required to respond to the challenge within a certain minimum time period. It has the objective to prevent the construction of a duplicate that releases a response within a certain time period, even though a module that releases it within a much longer period must be possible.

4. A NEW SECURITY PRIMITIVE: CRYPTOSTORAGE

4.1 Description of the Primitive

The security mechanism of "cryptostorage" first introduced in Section 3.1.2 is of fundamental importance. It can be characterized in the following manner:

- **Definition of the cryptostorage mechanism (4)**
  A clearstring is a string of bits physically stored in an information-storage system with no security mech-

anism. A cryptostorage mechanism is a s-challenge-dependent mapping from a clearstring to the storage module of a PUF. Its security objective is that an attacker finds no mechanism for the extraction of the clearstring from the PUF without the set of s-challenges. The s-challenge is a string of bits that designates a subset of the address space of the information-storage system.

When one compares this definition with the one of a cryptographic encryption algorithm (3), the close analogy between the two is obvious: the s-challenge is equivalent to the key, the storage system of the PUF to the cryptogram and the clearstring to the cleartext. If cryptographic encryption is the science of concealed (crypto) writing (graphein), research and development on PUFs can be identified as the science of concealed storage, or "cryptostorage". Cryptostorage protects the physical information[4] itself. Cryptographic encryption protects the meaning of the information[5] in the cryptogram. The close parallel between cryptographic encryption and cryptostorage is illustrated in Table 1.

In cryptographic encryption, principles and comprehensible systematic arguments (not necessarily proofs) based on mathematics and informatics ensure that the cleartext cannot be extracted without the key. In an analogous manner, in cryptostorage, principles based on physics and electrical engineering must ensure that the stored information cannot be extracted without being in possession of the s-challenges. The analog of cryptanalytic analyses of cryptographic encryption are, e.g., attacks using various forms of sophisticated learning programs[18].

The specific value of cryptostorage for embedded security is the systematic practical and theoretical development of information-storage systems that remain secure when the attacker can directly and completely access them, i.e. uses their regular read-out system. It will be necessary to develop the primitive of cryptostorage in a systematic manner because it will always remain impossible in principle to completely deny access to embedded storage systems. The major aim of cryptostorage will be to develop PUFs with a comprehensible, well understood security level based on sound principles of engineering, physics and mathematics.

5. FUTURE PUF RESEARCH

The two most basic question are:
1. What new security objectives for PUFs could there be? Which PUF storage-functionalities, besides the simple, addressed and timed release of information could be useful?
2. What new security mechanisms, besides the CS, NC and cryptostorage classes could there be?
The cryptostorage security primitive is bound to play a similarly important role in embedded security as cryptographic encryption does. Many proposed PUFs are based on it, but the security of these proposals still need further study. The foundations of cryptostorage need to be developed systematically: Which further standard design principles for physical memories could be given up, to make them security-memory bounded? How can EUR PUFs be constructed which are not based on quantum-storage? How can it be ensured that

---

[4]Physical information is a physical system that has a certain computable relation to the outside world in space an time.
[5]The meaning of information is a description of the relation between physical information and outside world.

Cryptographic encryption	Cryptostorage
1. meaning of Information	physical Information
2. cryptogram	PUF storage module
3. crypto-algorithm	cryptostorage mechanism
4. inversion of cryptogram	reproduction of PUF
5. cryptographic key	s-challenge
6. encrypt with key	store at s-challenge adr.
7. decrypt with key	apply s-chall. & measure

Table 1: A listing of corresponding concepts of cryptographic encryption and cryptostorage. 1. What is protected? 2. Where is protected information? 3. What is the security mechanism that 4. prevents what?

MRT PUFs with N secret elements require on the order of 2^N challenge-response pairs to extract all contained information even under sophisticated learning attacks? Another important research topic are algorithms and protocols to support PUF-based applications.

If a new "PUF protocol", e.g., for authentication is proposed, it should be communicated which of its features are PUF specific, and why. Otherwise, it would be a general authentication protocol, i.e., its novelty cannot lie in its applicability to PUFs.

6. CERTIFICATION OF PUFS

A first step in a PUF security evaluation must be to ascertain that a hardware element within a target of evaluation (TOE) is really a PUF according to our proposed definition. The next step will be to identify the PUF's security objectives within the TOE's architecture and the security mechanisms that fulfill these objectives. The central task of a PUF certification will be to theoretically model and verify the PUF's security mechanism in simulation and practice. Possible effects of the PUF's security mechanism on other conventional, non-PUF mechanism that secure the protected memory also need to be considered.

Classifying the PUF according to the characteristics in Section 3 will be useful to lay out the scope of the evaluation. Precisely defined security objectives clearly determine the tasks of the security mechanisms. If, e.g., the PUF has an internally generated random value as response (CSP mechanism), the machine producing it assumes the role of a Random-Number Generator (RNG). It will then be necessary to evaluate the manufacturing system, e.g., according to the existing guidelines for the certification of RNGs [19]. PUFs can be valuable because they reach new levels of security, but also because they allow cheaper solutions than conventional approaches. Therefore, security evaluations of PUFs must not just address whether a PUF can be reproduced but also at which effort.

7. CONCLUSION

We proposed to define PUFs as physical memories designed with the objective to protect a well defined storage functionality against duplication with a mechanism that is inseparable from the information-storage mechanism. PUFs

in this sense seem to completely characterize what the community always meant by this concept.

We welcome challenges to this claim.

Our definition does not restrict PUF architectures in any way, but provides a clear separation to other conventional security modules. Conventional secure memories in which information-storage and security mechanism are separable, i.e. protect against access to the informations-storage system, are no PUFs. In principle, their security mechanism can also protect entities different from associated memories. Therefore, a classification of such architectures as two independent security and storage architectures is more appropriate.

We argue that PUF research and development is a much more fundamental and important field than previously thought. In a sense, it is the discipline of the secure storage of physical information itself, just as cryptography is the field of secure writing of the meaning of information.

Acknowledgment

8. REFERENCES

[1] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2002, pp. 148–160.

[2] ——, "Delay-based circuit authentication and applications," in *Symposium on Applied Computing (SAC)*, 2003.

[3] R. Maes and I. Verbauwhede, "A discussion on the properties of physically unclonable functions," TRUST 2010 Workshop on Security Hardware, Berlin, DE, 2010.

[4] A.-R. Sadeghi, I. Visconti, and C. Wachsmann, "PUF-enhanced rfid security and privacy," in *Workshop on Secure Component and System Identification (SECSI)*, 2010.

[5] F. Armknecht, R. Maes, A.-R. Sadeghi, F.-X. Standaert, and C. Wachsmann, "A formalization of the security features of physical functions," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, ser. SP '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 397–412. [Online]. Available: http://dx.doi.org/10.1109/SP.2011.10

[6] U. Rührmair, H. Busch, and S. Katzenbeisser, "Strong PUFs: Models, constructions, and security proofs," in *Towards Hardware-Intrinsic Security*, A.-R. Sadeghi and D. Naccache, Eds. Springer, 2010, pp. 79–96.

[7] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," Cryptology ePrint Archive, Report 2011/657, 2011, http://eprint.iacr.org/2011/657.

[8] R. Plaga and F. Koob, "A formal definition and a new security mechanism of physical unclonable functions," in *Proceedings of the 16th international GI/ITG conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, ser. MMB'12/DFT'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 288–301.

[9] U. Rührmair, "Simpl systems: On a public key variant of physical unclonable functions," Cryptology ePrint Archive, International Association for Cryptologic Research, Tech. Rep., 2009.

[10] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Information Hiding*. Springer, 2009, pp. 206–220.

[11] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, pp. 9–14, 2007.

[12] B. Gassend, M. van Dijk, D. Clarke, E. Torlak, P. Tuyls, and S. Devadas, "Controlled physical random functions and applications," *ACM Transactions on Information and System Security*, vol. 10, no. 4, Jan. 2008.

[13] U. Rührmair, J. Sölter, and F. Sehnke, "On the foundations of physical unclonable functions," Cryptology ePrint Archive, Report 2009/277, 2009, http://eprint.iacr.org/.

[14] R. Maes, A. V. Herrewege, and I. Verbauwhede, "Pufky: A fully functional puf-based cryptographic key generator," in *CHES*, vol. 7428. Springer, 2012, pp. 302–319.

[15] F. Pastawski, N. Y. Yao, L. Jiang, M. D. Lukin, and J. I. Cirac, "Unforgeable noise-tolerant quantum tokens," 2011.

[16] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *9th International Workshop of Cryptographic Hardware and Embedded Systems, CHES'07*, 2007, pp. 63–80.

[17] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 10, pp. 1200–1205, December 2005.

[18] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM conference on Computer and communications security*, ser. CCS '10. New York, NY, USA: ACM, 2010, pp. 237–249.

[19] W. S. Wolfgang Killmann, "Ais 31 v3 - functionality classes and evaluation methodology for random number generators," Federal Office for Information Security (BSI), Germany, 2013, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizie /Interpretationen/AIS_31_pdf.pdf.