Protection Profile for PUF-Based Devices

Andrea Kolberger, Ingrid Schaumüller-Bichl, Verena Brunner, Martin Deutschmann

Protection Profile for PUF-based Devices

Andrea Kolberger[1], Ingrid Schaumüller-Bichl[1], Verena Brunner[2], and Martin Deutschmann[2]

[1] University of Applied Sciences Upper Austria
Department Secure Information Systems, Softwarepark 11, A-4231 Hagenberg
{andrea.kolberger,ingrid.schaumueller-bichl}@fh-hagenberg.at
[2] Technikon Forschungs- und Planungsgesellschaft mbH
Burgplatz 3a, A-9500 Villach
codes@technikon.com

Abstract. Physically Unclonable Functions (PUFs) are a promising technology in cryptographic application areas. The idea of PUFs is to make use of the unique "fingerprint" of the IC, to enable generation of secrets or keys without storing sensitive data permanently in memory. Since PUFs are "noisy" functions, some kind of post processing is required to reliably reconstruct the respective PUF response. Based on potential threats and vulnerabilities as well as the security requirements for PUF-based tokens we developed a draft version of a Protection Profile according to Common Criteria. This paper discusses the central parts of this Protection Profile, namely the Target of Evaluation (TOE), PUF-specific security functional requirements (SFRs), and requirements on the operational environment regarding the whole life cycle of the TOE.

1 Introduction

The security of IT systems in various domains, such as consumer electronics, automotive, avionics or control systems is gaining in importance more and more. On the other hand nobody is willing to pay extra for a higher level of security. Therefore, designers of security solutions have to build systems, which offer a reasonable level of security, while being economically attractive. The technology of Physically Unclonable Functions (PUFs) [8] promises to be a good candidate to serve both demands, as the idea can be realized cost-effectively, while still providing a higher level of security than state-of-the-art solutions.

PUFs are special challenge-response entities, which make use of intrinsic variations in the integrated circuit (IC) - which are out of the control of the manufacturer - to build cryptographic applications such as secure key storage or authentication protocols. The technology is explained in more detail in section 2. In order to collect and formalize the requirements for PUF-based systems we prepared a draft version of a Protection Profile (PP) according to Common Criteria (CC). Thus the present paper targets the crypto community, dealing with PUF applications and industry partners considering PUFs as a potential technology. Section 3 gives a brief overview of the CC and some information regarding PPs whereas section 4 describes the intended TOE and its components. The PUF-specific requirements defined and selected from CC to ensure a secure operation of the TOE are summarized in section 5. The paper comes to an end with a conclusion and outlook on our ongoing work in section 6.

2 PUF Technologies

One basic principle of cryptographic applications is that the security of a system relies on the secrecy of the used key (Kerckhoff's principle). Therefore the question where and how to store a secret key is essential for the level of security a system can provide. The usage of Physically Unclonable Functions (PUFs) allows designing cryptographic systems, where the key is not present in memory at all. Only the combination of some non-sensitive information, called helper data, and the intrinsic, unclonable properties of the PUF instantiation allows the reconstruction of the key.

The basic principle of PUFs is to exploit unique information which originates from submicron variations in the manufacturing process in integrated circuits (ICs), which are out of the control of the manufacturer. One established technique is to utilize the start-up behaviours of SRAM cells to serve as the digital fingerprint. If we view each SRAM cell as a single bit, the resulting bit string will tend to have the same value every time the device is powered, however, owing to time, temperature variations and voltage ramp-up variations, some bits tend to flip [9]. Another widespread PUF instantiation is the so-called Arbiter PUF, which belongs to the group of delay-based intrinsic PUFs [7]. The basic idea is to conduct a race on two paths on a chip and then let an Arbiter circuit decide, which path "won" the race. The challenges consist of a vector shaping the path of the "race". For further PUF constructions we refer to Maes and Verbauwhede [11] who present an extensive overview of all PUFs and PUF-like proposals.

Responses generated by PUFs are noisy by nature, i.e. when a single PUF is challenged with one and the same challenge several times it always returns a slightly different response. Such responses cannot be directly used for cryptographic applications. Thus post processing methods are used to reliably produce and reproduce a certain PUF response or to derive a cryptographic key. The post processing includes a procedure to perform information reconciliation which is mostly realized by some error correcting algorithm and a procedure to extract nearly uniform randomness, which can be realized with help of cryptographic hash function. The combination of these two procedures is called a fuzzy extractor (FE). For more information about FE realizations, we refer to [5].

PUF instantiations are amongst others characterized by the so-called inter- and intra-distance. For a particular challenge, the inter-distance between two PUF instantiations is the distance between the two responses resulting from applying this challenge once to both PUFs. On the other hand, we define for a particular challenge, the intra-distance between two evaluations on one single PUF instantiation as the distance between two responses resulting from applying this challenge twice to one PUF.

3 Common Criteria and Protection Profiles

The Common Criteria for Information Technology Security Evaluation (CC) are internationally accepted criteria to evaluate the security functionality of a product and the correctness of its design and implementation (assurance). Part 1 of Common Criteria [1] provides an introduction to CC and describes the general model. Requirements that might be fulfilled by a certain product are specified in Part 2 [2] and 3 [3]. Part 2 contains requirements intended to provide the security functionality of a product, the so-called security functional requirements (SFRs). In order to be able to evaluate the security functionality provided by a product the CC have defined security assurance requirements (SARs) in Part 3.

The central documents of CC are the Protection Profile (PP) and Security Target (ST). Protection Profiles describe the SFRs and SARs for a product class, i.e. a PP describes which requirements a certain product has to fulfill but it is not defined how these requirements are implemented. In comparison to that a Security Target contains security requirements for a specific product and defines how these requirements are implemented. The development and evaluation of a Security Target is mandatory for certification issues against CC.

4 Protection Profile for PUFs - the TOE

In the CODES project[1] we worked out a draft version of a Protection Profile for Physically Unclonable Functions (PUFs). Our PP contains all required parts, however this paper is confined to the TOE definition, the security functional requirements and the requirements on the operational environment of the TOE.

Figure 1 provides an overview of the TOE design that is intended to realize two use cases, namely *Mutual Authentication* [10] and *Secret Key Generation/Session Key Exchange* [6]. In the authentication process both verifier and PUF-based device are capable to (re-)construct a PUF response and verify the authenticity of each other. The second use case is intended to encrypt the initial communication between two entities with an error corrected PUF response (common secret, symmetric key) agreed upon in the enrolment phase. Part of the encrypted messages is a session key that is used for further communication.

The **pre-operational environment** includes procedures that are performed during the development process of the TOE. One major part in this environment is the initialization, personalization and enrolment of the TOE where data like the ID of the TOE, challenge-response pairs (CRPs) and helper data are generated and stored. This information is unique and essential for the security functionality of the TOE (TSF). The **operational environment** contains components like database and terminals that will be combined with the TOE in the composite product integration. The **TOE** itself shall be implemented on one single IC. Below, the components providing the security functionality are

[1] Project CODES: Research activities are on post processing methods like error correction codes and anti-ageing techniques to raise the stability of PUF-based responses and thereby the reliability of PUF-based security modules.

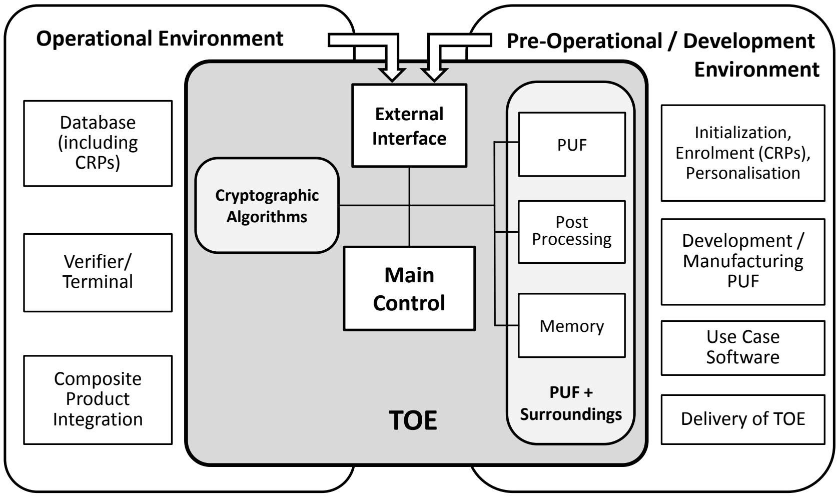

Operational Environment
Pre-Operational / Development Environment
Database (including CRPs)
Verifier/ Terminal
Composite Product Integration
External Interface
Cryptographic Algorithms
Main Control
TOE
PUF
Post Processing
Memory
PUF + Surroundings
Initialization, Enrolment (CRPs), Personalisation
Development / Manufacturing PUF
Use Case Software
Delivery of TOE

Fig. 1. TOE with its surrounding environment within the whole life cycle

briefly described. The **main control** is the central component which controls communication between and access to all other TOE components. In order to communicate with the TOE's environment an **external interface** needs to be implemented. The **PUF** type itself is not specified in this TOE. Depending on the use case, either a memory- or delay-based PUF can be used. The PUF can only be accessed by the main control which then returns a response that will directly serve as an input to the post processing. Due to the fact that PUFs are noisy functions some post processing procedure is necessary in order to correct occurred errors and reconstruct certain information. Depending on the intended use case the post processing can be realized by one of the following features:

- **Secure Sketch:** The reproduction step of the secure sketch uses helper data to reconstruct the original PUF response. The helper data is regarded non-sensitive, as it provides only negligible information about the actual secret.
- **Fuzzy Extractor:** The reconstruction step of the fuzzy extractor uses helper data generated in the pre-operational environment to reproduce a secret that has also been enrolled in the pre-operational environment.

Memory is used to store some information which is necessary to provide the TOE's security functionality. These data might be helper data or the ID of the TOE. The protection of the memory space of the IC against attacks is not in the scope of this PP. **Cryptographic functions** will be necessary in order to realize the complete use case. In case of key generation it is necessary that the respective symmetric or asymmetric cipher is available on the TOE in order to be able to en- or decrypt information.

5 Security Requirements and Extended Components

A risk analysis on the use cases described in section 4 highlighted the main threats that have to be countered for PUF-based devices and showed that the usage of a weak fuzzy extractor or secure sketch as well as PUF failures cause the highest risks. As a consequence the developed post processing methods must not reveal any information regarding the PUF response. At the same time these methods have to reliably reconstruct secrets from an errorprone PUF response.

5.1 Security Functional Requirements

Our PP includes numerous security functional requirements concerning detection of and reaction on malicious activities, control of the internal workflow, and selftests to provide an initial secure start-up of the TOE. Below we concentrate on SFRs [2] that focus on the specific needs of PUF-based security schemes.

PUF and Cryptographic Functions. The TSF challenges the PUF using a predefined challenge from the CRP database. Depending on the underlying PUF construction a PUF response is generated. According to the intended use case the PUF response is transferred between the TOE's components and might be directly used as a cryptographic key after post processing or it serves as an input to a cryptographic key generation algorithm. Therefore generated secrets shall meet defined quality metrics (e.g. number of required bits).

FPT_PUF.1 Physically unclonable function, requires that according to a challenge a PUF response is generated.
FCS_CKM.1 Cryptographic key generation, requires cryptographic keys to be generated in accordance with a specified algorithm and key sizes which can be based on an assigned standard.
FCS_COP.1 Cryptographic operation, requires a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard.
FIA_SOS.2 TSF Generation of secrets, requires the TSF to be able to generate secrets that meet defined quality metrics.

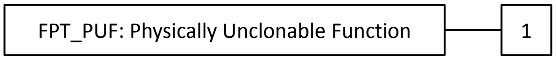
Reliable Post Processing Methods. Post processing methods are used to reconstruct secrets/keys from a noisy PUF response. The error correction mechanism shall be implemented in such a way that with the help of helper data a secret/key can be reconstructed reliably, i.e. it shall be capable to correct a certain number of errors even though of environmental variations or ageing effects of a PUF. Helper data extracted by the fuzzy extractor needs to be generated in the pre-operational environment and must not reveal any information about the response. Depending on the use case helper data are generated by the TOE itself, by an external entity or they are already stored in the TOE's memory.

FCS_CPP.1 Secure Sketch, requires helper data to reconstruct a PUF response.
FCS_CPP.2 Fuzzy extractor, requires helper data to reconstruct a cryptographic key or secret.

5.2 Extended Component Definition

PUF-based security schemes provide functionalities that are not based on components specified in CC Part 2. For the use of PUFs and the according post processing methods new families were defined.

Definition of the Family FPT_PUF. This family defines requirements that a PUF is used to derive a PUF response according to a challenge. The PUF response is subsequently used for authentication procedures or generation of cryptographic keys or secrets.



`FPT_PUF.1`	**Physically Unclonable Function**
Hierarchical to:	No other components.
Dependencies:	[`FCS_CPP.1` Secure Sketch or `FCS_CPP.2` Fuzzy Extractor]
`FPT_PUF.1.1`	The TSF shall provide a capability to challenge a [assignment: *list of PUF types*] in order to receive a PUF response.

Definition of the Family FCS_CPP. This family defines requirements for the post processing step necessary for PUF-based applications. Depending on the use case, helper data is used either in combination with a secure sketch to reproduce a PUF response or in combination with a fuzzy extractor to reconstruct a cryptographic key or secret.

FCS_CPP: Cryptographic post processing
1
2

`FCS_CPP.1`	**Secure Sketch**
Hierarchical to:	No other components.
Dependencies:	`FPT_PUF.1` Physically Unclonable Function
`FCS_CPP.1.1`	The TSF shall provide a capability to reproduce a PUF response with the help of helper data received from [selection: *list of entities that extract or store helper data*].
`FCS_CPP.2`	**Fuzzy Extractor**
Hierarchical to:	No other components.
Dependencies:	`FPT_PUF.1` Physically Unclonable Function
`FCS_CPP.2.1`	The TSF shall provide a capability to reproduce a cryptographic key or secret with the help of helper data received from [selection: *list of entities that extract or store helper data*].

In addition to the two families described above we also defined extended components concerning limited capability (`FMT_LIM.1`) and limited availability (`FMT_LIM.2`) of functions [7].

5.3 Requirements and Assumptions on the Operational Environment

Requirements defined in this section are addressed to the TOE's operational environment. That implies requirements which cannot be fulfilled by the TOE's security functionality and therefore have to be realized by the operator/enduser to ensure the secure, correct and effective operation of the TOE. The most important requirements and assumptions are described below.

Manufacturing, Enrolment and Delivery of PUF-based tokens. In the contract concluded between the manufacturer of the TOE and operator has to be stated that the manufacturer is not allowed to add any functionality that enables prediction or specific manipulation of the PUF response anyway. Further the operator should be capable to verify the correct implementation of the PUF. In the enrolment phase the database, including CRPs as well as common secrets or helper data for a specific TOE uniquely identified by an ID, shall be generated in a secure manner and secure environment. The database including PUF-specific and consequently confidential data has to be transferred in a secure manner between the enrolment facility and the customer. Therefore authenticity, confidentiality and integrity of the stored data has to be ensured. Further the enroller has to make sure that only the intended person receives the CRP database.

Selection of Challenges. Enrolment shall take place in secure environment as mentioned before. During enrolment a PUF shall be challenged with randomly chosen and unpredictable challenges. Further the challenges for different PUF-based token must not be equal and should be made of a sufficient length in order to make brute force attacks harder or inefficient.

Temperature/Voltage. It is assumed that the temperature of the TOE's operational environment is within the range of $-40\ ^{\circ}C$ up to $+85\ ^{\circ}C$ [12]. Regarding the variation of voltage PUF technologies react differently. Therefore an assumption on the range of voltage, within that a certain PUF construction works reliable, has to be made in the ST. Thus the operator shall ensure that the temperature of the TOE's operational environment and the variation of voltage is within the specified range otherwise the TOE might not work reliably.

6 Conclusion and Outlook

To the best of our knowledge no Protection Profile for the technology of PUFs exists at the moment. Our draft version is going to be followed up in the ongoing project. Especially the requirements defined so far might be refined, replaced or some further security functionality might be added. Since the goal of the project is not to certify a "real world" TOE, our draft PP is not object to evaluation activities in order to judge its suitability. The aim is to elaborate a Protection Profile for PUFs that might form the basis for future work on PPs to enable certification of PUF-based devices against Common Criteria.

Acknowledgements. This work is co-financed by the Austrian Research Promotion Agency (FFG) in the FIT-IT line within the project CODES (835932): Algorithmic extraction and error correction codes for lightweight security anchors with reconfigurable PUFs. The project partners are Technikon Forschungs- und Planungsgesellschaft mbH, Alpen-Adria Universität Klagenfurt and University of Applied Sciences Upper Austria - Campus Hagenberg.

References

1. Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model. CCMB-2012-09-001, Version 3.1, Revision 4 (September 2012)
2. Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components. CCMB-2012-09-002, Version 3.1, Revision 4 (September 2012)
3. Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components. CCMB-2012-09-003, Version 3.1, Revision 4 (September 2012)
4. Eurosmart Security IC Platform Protection Profile. Version 1.0, BSI-PP-0035 (June 2007)
5. Bösch, Ch., Guajardo, J., Sadeghi, A.-R., Shokrollahi, J., Tuyls, P.: Efficient Helper Data Key Extractor on FPGAs, In Elisabeth Oswald and Pankaj Rohtagi, editors, Cryptographic Hardware and Embedded Systems CHES 2008. LNCS, vol. 5154, pp. 181–197. Springer, Heidelberg (2008)
6. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM Journal on Computing, 38(1):97–139 (2008)
7. Fruhashi, K., Shiozaki M., Fukushima A., Murayama T., Fujino T.: The arbiter-PUF with high uniqueness utilizing novel arbiter circuit with Delay-Time Measurement. IEEE International Symposium on Circuits and Systems (ISCAS) 2011, pp. 2325–2328 (2011)
8. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Controlled Physical Random Functions. In IEEE, editor, Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC02). USA, (2002)
9. Handschuh, H.: Hardware-Anchored Security Based on SRAM PUFs, Part 1. Security Privacy, IEEE, 10(3):80–83 (May/June 2012)
10. Herrewege, A., Katzenbeisser, S., Maes, R., Peeters, R., Sadeghi, A.-R., Verbauwhede, I., Wachsmann, Ch.: Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs. In AngelosD. Keromytis, editor, Financial Cryptography and Data Security. LNCS, vol. 7397, pp. 374–389. Springer, Heidelberg (2012)
11. Maes, R., Verbauwhede, I.: Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions. In Ahmad-Reza Sadeghi and David Naccache, editors, Towards Hardware-Intrinsic Security, Information Security and Cryptography, pp. 3–37. Springer, Heidelberg (2010)
12. Schrijen, G.-J. and van der Leest, V.: Comparative analysis of SRAM memories used as PUF primitives. Design, Automation Test in Europe Conference Exhibition (DATE), 1319–1324 (2012)