Quality Limitations on the Extraction of a PUF-based Cryptographic Key

Sandra L. Lattacher, TECHNIKON Forschungs- und Planungsgesellschaft mbH
Joint work with Martin Deutschmann, Michael Höberl, Christina Petschnigg, and Naeim Safari

1. Introduction

An indispensable demand for the majority of cryptographic implementations is the ability to securely generate and store cryptographic keys. Physically Unclonable Functions (PUFs) prove to be a suitable primitive to comply these requirements. PUFs can be understood as physical systems which, when measured, provide unique and unpredictable responses. The responses are depending on the physical structure of the device and are out of the control of the manufacturer. As PUFs are physical objects they are prone to errors, i.e. the responses will always include some degree of noise. Further the distribution of the responses does not necessarily have to be uniform. In other words, when designing key generation or protection schemes based on PUF measurements, one has to make sure that suitable error correcting mechanisms are put in place. Further entropy extraction is crucial, to ensure the generation of keys with high entropy.

In this work we are exposing the limits in the design of a key generation framework by taking into account relevant properties of PUF instantiations, such as entropy or the mutual information. Our attempt is to present a tool to evaluate if and how a cryptographic key of a certain length can be extracted with the demanded reliability for a given PUF source. The solutions presented are evaluated against concrete PUF parameters. The PUF source are 65nm TSMC ASICs, which were developed in the course of the FP7 research project UNIQUE [1].

2. The PUF Framework

By challenging the PUF twice consecutively with identical conditions, we expect an unreliability of the two responses R and R' reflected by the so-called intra distance. Moreover, when comparing two different instances of a PUF type, an inter distance of about 50% is desired.

To reliably generate cryptographic keys, the PUF response has to be processed within a specific framework that can cope with the noise cancellation and the entropy extraction. This is where so called Helper Data Algorithms (HDAs) come into play. Most HDAs follow a two-step approach: the key is derived by querying the PUF in a secure environment during an enrollment phase. During the so called reconstruction phase the key is recovered in the field. A HDA can additionally be divided into three sub-components. The first is bit selection, aiming at discarding the least reliable bits within a PUF response. This step can significantly lower the number of expected errors within the response and thus allowing the application of shorter and simpler error correcting codes. Applying bit selection as well as error correction measures allow the assumption of a negligible low failure rate during reconstruction. However, the remaining bits have non-maximum entropy due to leakage during the former two steps. Therefore, the third step comprises entropy compression. [5]

3. Quality Aspects of a Key

Generally speaking there are two main areas that might affect the quality level of a key: the property of the raw data, and the helper data leakage, including the choice of the error correction and the randomness extraction. A rough assessment on the quality of the data is the Hamming weight of the response which gives a first indication of the randomness, since it is a measure of the distribution of ones and zeros within a binary bit string:

[1] www.unique-project.eu

$$W(x) = \sum_{i=1}^{n} x_{\mathrm{i}}. \tag{1}$$

When designing key generation frameworks, a high level of unpredictability and robustness is claimed. Entropy estimation, which is the measure of uncertainty of a random variable, comprises in fact all relevant parameters.

The Shannon entropy which is commonly used in information theory is defined as

$$H_1(x) = -\sum_{i=1}^{n} p_{\mathrm{i}} \log_2 p_{\mathrm{i}}, \tag{2}$$

where x defines the binary random variable and p_{i} the probability that x takes on zero or one. The limit of H_1 converges to the min-entropy:

$$H_\infty(x) = -\log \max_i p_{\mathrm{i}}. \tag{3}$$

When transmitting information, it is assumed to be correct on the receiver side, but in fact, the signal will be superimposed by noise with a specific bit error probability p_b. Assuming a given bit error probability we claim a failure rate of $P_{\mathrm{fail}} \leq 10^{-6}$. For simple codes, an estimation of the probability that a string of n bits has more than t errors is given by:

$$P_{\mathrm{fail}} = \sum_{i=t+1}^{n} \binom{n}{i} p_b^i (1-p_b)^{n-i} \tag{4}$$

With the use of a HDA, a key is derived from the raw PUF source bits by compressing the bits with a hash function. The amount of source bits that are needed to achieve a secret of a specific size is expressed in the so-called secrecy rate. The maximum achievable secrecy rate depends on the mutual information

$$I(x,y) = H(x) - H(x|y) \tag{5}$$

between the measurement done at enrollment x and reconstruction y, where $H(x|y)$ describes the remaining entropy of x when y is known. In more detail, $\lceil K/I(x,y) \rceil$ gives the number of required source bits to derive a secret of size K. [8] [18]

Given any $C[n,k,d]$ code the entropy loss within a practical realization of a HDA can be stated as $n-k$. It follows that the leftover entropy ℓ in the PUF response is given by $\ell = m + k - n$, relying on the commonly used Random Oracle model [2] [12], where the loss during the entropy extraction is assumed as 0.

4. Evaluation and Limitations

In the following, our aim is to show the limitations and boundary conditions when generating a 128-bit binary key based on different PUF instantiations, when having the quality aspects of Section 3 in mind. The used ASICs containing different PUF types are mounted on a customized evaluation board, that is connected via a ribbon cable to a Xilinx KC705 FPGA evaluation board. The ASICs are controlled via a dedicated IP block on the FPGA and the measurements are forwarded via a serial interface to a PC. We focused during the evaluation of raw PUF data on memory based PUFs, namely SRAM, Latch and DFF PUFs.
Table 1 lists in addition to the bit error rate, also the maximum number of source bits N as well as the entropy $H_1(x)$, the min-entropy $H_\infty(x)$, the mutual information $I(x,y)$ and the Hamming weight $W(x)$ for SRAM, DFF and Latch PUFs. The SRAM PUF behaves worst regarding the bit error rate with a value of 5.2%. In contrast, the other parameters such as the entropy values or the

Type	p_b	N	$H_1(x)$	H_∞	$I(x,y)$	$W(x)$
SRAM	5.2%	65536	1.00	0.99	0.70	0.49
DFF	3.1%	8192	0.84	0.45	0.64	0.73
Latch	2.5%	8192	0.79	0.39	0.63	0.76

TABLE 1. Quality measures of memory-based PUFs at room temperature.

Hamming weight come close to an optimum. The biased output of DFF and the Latch PUF expressed by the Hamming weight influences the entropy and the min-entropy, which is significantly lower for these two PUF types. Figure 1 depicts the dependency between the code parameters n and d with respect to a failure rate $P_{\text{fail}} \leq 10^{-6}$ for the evaluated PUF types.

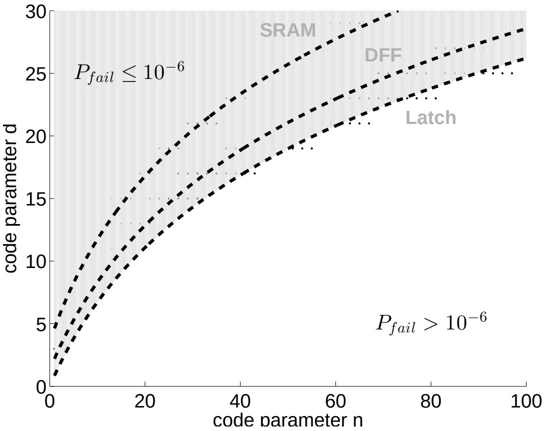

FIGURE 1. Dependency of the code parameters n and d regarding the failure rate P_{fail} for SRAM, DFF and Latch PUFs.

Targeting secure key generation, the leftover entropy is the second point that has to be considered. It is common practice to process the PUF response in blocks of equal length n. With a given entropy, the mutual information and a desired key length of 128 bit, a lower bound of the block numbers l_{min} is derived. Depending on the number of available source bits N, an upper bound l_{max} can be determined. As long as $l_{\text{max}} > l_{\text{min}}$, the key generation will be successful with maximum achievable entropy. Based on these assumptions, all free parameters can be combined to a threshold function

$$f_T(k,n) = k - sn - o, \tag{6}$$

where s is the slope, o the offset of the function and k, n are the variable code parameters. In Table 2 the fixed parameters of f_T are shown for a changing number of source bits and for the different PUF types. Generally, the slope of the threshold function tends to increase, when the entropy decreases at the same time. The key can be derived with maximum achievable entropy with a specific code when $f(k,n) \geq 0$.

5. CONCLUSION

The aim of this paper was to present a hands-on guide on how to design tailored PUF-based key generation frameworks, taking into account the limitations given by the PUF source and the HDA. With the implementation of a threshold function, we are able to expose the limits of reliable key generation while considering the relevant quality aspects at the same time. There are a couple of published papers that describe the design of HDAs, the choice of the error correcting code, the

	1024		2048		4096		8192	
Type	s	o	s	o	s	o	s	o
SRAM	0.19	0.32	0.09	0.46	0.04	0.55	0.02	0.59
DFF	0.68	0.39	0.61	0.50	0.58	0.47	0.57	0.50
Latch	0.74	0.43	0.67	0.53	0.64	0.48	0.63	0.56

Table 2. Parameters of the threshold function for the feasibility of key generation given the slope s and the offset o.

consideration of entropy loss or statistical analysis of different PUF sources. To our best knowledge, however, there is no paper that tries to draw a complete picture, reaching from statistical investigation on the PUF source to the actual HDA design.

Acknowledgment. This work was supported in part by the European Commission through the ICT programme under contract FP7-ICT-2011-317930 HINT.

References

[1] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Berk Sunar, and Pim Tuyls, *Memory leakage-resilient encryption based on physically unclonable functions*, Towards Hardware-Intrinsic Security, Springer, 2010, pp. 135–164.

[2] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, Francois-Xavier Standaert, and Yu Yu, *Leftover hash lemma, revisited*, Cryptology ePrint Archive, Report 2011/088, 2011, `http://eprint.iacr.org/`.

[3] Christoph Boesch, Jorge Guajardo, Ahmad-Reza Sadeghi, Jamshid Shokrollahi, and Pim Tuyls, *Efficient Helper Data Key Extractor on FPGAs*, Proceedings of the 10^{th} International Workshop on Cryptographic Hardware and Embedded Systems CHES 2008, Lecture Notes in Computer Science, vol. 5154, Springer Berlin Heidelberg, 2008, pp. 181–197 (English).

[4] J. Delvaux and I. Verbauwhede, *Key-recovery attacks on various RO PUF constructions via helper data manipulation*, Proceedings of the Conference on Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014, March 2014, pp. 1–6.

[5] Jeroen Delvaux, Dawu Gu, Dries Schellekens, and Ingrid Verbauwhede, *Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **PP** (2014), no. 99, 1–1.

[6] Jeroen Delvaux and Ingrid Verbauwhede, *Fault Injection Modeling Attacks on 65nm Arbiter and RO Sum PUFs via Environmental Changes*, IACR Cryptology ePrint Archive **2013** (2013), 619.

[7] Jeroen Delvaux and Ingrid Verbauwhede, *Attacking PUF-Based Pattern Matching Key Generators via Helper Data Manipulation*, Topics in Cryptology CT-RSA 2014, Lecture Notes in Computer Science, vol. 8366, Springer International Publishing, 2014, pp. 106–131 (English).

[8] Jorge Guajardo, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls, *FPGA Intrinsic PUFs and Their Use for IP Protection*, Proceedings of the 9^{th} International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2007, Lecture Notes in Computer Science, vol. 4727, Springer Berlin Heidelberg, 2007, pp. 63–80 (English).

[9] Maximilian Hofer and Christoph Boehm, *An Alternative to Error Correction for SRAM-Like PUFs*, Proceedings of the 12^{th} Internatinal Workshop on Cryptographic Hardware and Embedded Systems, CHES 2010, Lecture Notes in Computer Science, vol. 6225, Springer Berlin / Heidelberg, 2011, pp. 335–350.

[10] Stefan Katzenbeisser, Ünal Kocabaş, Vladimir Rožić, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann, *PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions(PUFs) Cast in Silicon*, Proceedings of the 14^{th} Internatinal Workshop on Cryptographic Hardware and Embedded Systems CHES 2012, Lecture Notes in Computer Science, vol. 7428, Springer Berlin Heidelberg, 2012, pp. 283–301 (English).

[11] P. Koeberl, Jiangtao Li, A. Rajan, and Wei Wu, *Entropy loss in puf-based key generation schemes: The repetition code pitfall*, Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on, May 2014, pp. 44–49.

[12] Roel Maes, *Physically unclonable functions - constructions, properties and applications*, Springer, 2013.

[13] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede, *Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs*, Proceedings of the 11^{th} International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2009, Lecture Notes in Computer Science, vol. 5747, Springer Berlin Heidelberg, 2009, pp. 332–347 (English).

[14] Roel Maes, Anthony Van Herrewege, and Ingrid Verbauwhede, *PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator*, Proceedings of the 14^{th} International Workshop on Cryptographic Hardware and Embedded Systems CHES 2012, Lecture Notes in Computer Science, vol. 7428, Springer Berlin Heidelberg, 2012, pp. 302–319 (English).

[15] Roel Maes and Ingrid Verbauwhede, *Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions*, Towards Hardware-Intrinsic Security, Springer Berlin Heidelberg, 2010, pp. 3–37 (English).

[16] R.S. Pappu, *Physical one-way functions*, Massachusetts Institut of Technology, 2001.

[17] Ying Su, J. Holleman, and B.P. Otis, *A Digital 1.6 pJ/bit Chip Identification Circuit Using Process Variations*, IEEE Journal of Solid-State Circuits **43** (2008), no. 1, 69–77.

[18] Robbert van den Berg, Boris Skoric, and Vincent van der Leest, *Bias-based Modeling and Entropy Analysis of PUFs*, Proceedings of the 3rd International Workshop on Trustworthy Embedded Devices (New York, NY, USA), TrustED '13, ACM, 2013, pp. 13–20.

[19] Vincent van der Leest, Bart Preneel, and Erik van der Sluis, *Soft Decision Error Correction for Compact Memory-Based PUFs Using a Single Enrollment*, Proceedings of the 14^{th} International Workshop on Cryptographic Hardware and Embedded SystemsCHES 2012, Lecture Notes in Computer Science, vol. 7428, Springer Berlin Heidelberg, 2012, pp. 268–282.