Noise Reduction on Memory-based PUFs

Mafalda Cortez Said Hamdioui

Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{A.M.M.O.Cortez, S.Hamdioui}@tudelft.nl

Vincent van der Leest Roel Maes Geert-Jan Schrijen

Intrinsic-ID B.V.
High Tech Campus 9,
Eindhoven, The Netherlands
{Vincent.van.der.Leest, Roel.Maes,
Geert.Jan.Schrijen}@intrinsic-id.com

***Abstract*—The efficiency and cost of silicon PUF-based applications, and in particular key generators, are heavily impacted by the level of reproducibility of the bare PUF responses under varying operational circumstances. Error-correcting codes can be used to achieve near-perfect reliability, but come at a high implementation cost especially when the underlying PUF is very noisy. When designing a PUF-based key generator, a more reliable PUF will result in a less complex ECC decoder and a smaller PUF footprint, hence an overall more efficient implementation. This paper proposes a novel insight and resulting technique for reducing noise on memory-based PUF responses, based on adapting supply voltage ramp-up time to ambient temperature. Circuit simulations on 45nm Low-Power CMOS are presented to validate the proposed methods. Our results demonstrate that choosing an appropriate voltage ramp-up for enrollment and adapting it according to the ambient temperature at key-reconstruction is a powerful method which makes memory-based PUF response noise up to three times smaller.**

I. Introduction

In recent years, silicon *Physically Unclonable Functions* (PUFs) [1] have been well established as innovative hardware security primitives. Numerous constructions have been proposed and implemented (see, e.g., [2] for an overview), and their interesting properties are being extensively investigated in large scale experiments [3–5]. A silicon PUF's ability to generate device-unique fingerprints based on deep-submicron silicon process variations makes it a highly practical tool for device identification. In addition, the intriguing and unparalleled property of *physical unclonability* is a strong foundation for deploying a silicon PUF as a security primitive.

Combined with proper post-processing, a PUF is able to generate secret keys of cryptographic strength [6,7], and reliably store them in a highly secure manner without the need for conventional on-chip *Non-Volatile Memory* (NVM). The key is derived from the device-intrinsic randomness which is evaluated by the silicon PUF. The main purpose of a PUF-based key generator is twofold: *i)* increasing the *reproducibility* of a typically noisy PUF evaluation to near-perfect reliability, and *ii)* accumulating sufficient *unpredictability* of possibly low-entropic PUF responses into a highly unpredictable cryptographic key. It is evident that the natural reproducibility and unpredictability of a bare silicon PUF implementation have a strong impact on the efficiency, and hence on the cost of a PUF-based key generator as a whole. A PUF with less noisy and more random responses will result in a key generator which requires less "PUF material", and hence less silicon area, to produce a reliable cryptographic key.

To produce a key with a practically acceptable reliability level (e.g., failure rate $\leq 10^{-6}$), a PUF-based key generator based on a fuzzy extractor [8,9] uses *Error-Correcting Codes* (ECC) to correct noisy PUF responses.These ECC techniques are very effective in boosting the reliability but tend to be computationally intensive. Moreover, the helper data, which is an unavoidable byproduct of the fuzzy extractor, will partially disclose the unpredictability of the bare PUF responses. This needs to be compensated for by using more PUF material and hence a larger PUF. Both the complexity of the ECC decoder, and the amount of randomness loss due to the helper data, scale with the required error correction capability of the ECC, i.e. less reliable PUF responses will result in a more complex decoder and a larger silicon PUF footprint. Hence, there is a strong incentive to use a PUF construction with an as high as possible reproducibility of its bare responses. This objective is seriously complicated by the reproducibility deterioration of silicon PUFs when subjected to varying operating conditions, like temperature and supply voltage variations.

Substantial research effort has been put into reliability enhancement of PUF-based key generators. Careful selection of the right ECC algorithms minimizes the helper data loss and decoder implementation cost [10,11]. On a physical level, construction improvements have been proposed to decrease the noise level of the bare silicon PUF responses directly, by modifying the PUF circuit [12,13] or the wafer mask set [14]. Analyzing a silicon PUF's susceptibility to its operating conditions has been explored for reliability enhancement [15,16].

In this work, we take this one step further by considering the combined effect of different operating parameters, in particular temperature and supply voltage ramp-up time, and their impact on the reproducibility of SRAM memory-based PUF responses. It is well known that temperature impacts the switching speed of electronic devices and contributes to electronic noise [3], whereas the voltage ramp-up time (i.e., the time it takes to reach the operational supply voltage after power-on) influences the power-up state of an SRAM [17,18]. This paper shows that intelligent matching of voltage ramp-up time to ambient temperature significantly improves the reproducibility of PUF responses at extreme temperatures, with noise levels up to 3× smaller than without matching. Moreover, this effective technique requires only a small number of additional building blocks and does not impose any modifications to the actual standard memory cell circuit. These effects are demonstrated in simulation for SRAM PUFs [6,17]. Industrial results for SRAM PUF and other memory-based

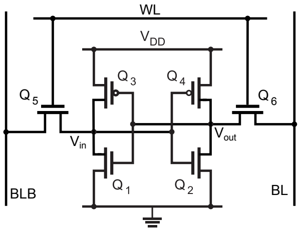

Fig. 1: SRAM Cell transistor level schematic.

PUF types [19–22] will be available by the time of publication.

The remainder of this paper is organized as follows. Section II provides a brief background on memory-based PUFs and PUF-based key storage. Section III discuses the simulation setup, including the noise metric, and the simulation results. The obtained results are discussed in more detail in Section IV, and Section V provides possible implementation options. Finally, Section VI concludes the paper.

II. Background: PUFs and Key Generation

This section first briefly provide some preliminaries on the basic operation of memory-based PUFs. Then, it shows how PUFs are deployed in a key storage system, and thereafter it gives the PUF's main quality metrics.

A. Memory-based PUFs

Memory-based PUFs [6,19–22] comprise bistable circuits, i.e., having two possible stable states denoted as logic '0' and '1'. Fig. 1 shows a typical six-transistor SRAM cell with at its core a basic bistable circuit consisting of two cross-coupled inverters, respectively formed by (Q_1, Q_3) and (Q_2, Q_4). The peripherical circuitry used to access the cell is comprised by two pass transistors (Q_5 and Q_6), the bitline, the complement bitline and the wordline. When powered-up, the cross-coupled inverters start driving electric current, hence increasing the voltages at their gates (V_{in} and V_{out}). The first inverter that builds enough gate voltage to drive its NMOS (i.e., NMOS V_{th}) will pull-down its output, forcing the other inverter to pull-up and causing the SRAM cell to setlle in one of both stable states. Since both inverters are designed to be nominally identical, the outcome (in which of both states a cell settles) is entirely determined by the effect of random process variations. An SRAM cell power-up state is hence in effect a PUF response, and the corresponding construction is called an SRAM PUF [6].

B. PUF-based Key Generation and Storage

Fig. 2 shows the basic flow of a PUF-based key generation and storage system [6,7] based on a fuzzy extractor [8,9], which typically consists of two phases:

(a) **Enrollment:** a key is generated from a *PUF Reference Response* (PRR) as shown in Fig. 2(a). First, the PUF is evaluated and produces the PRR. Next, the PRR is processed by the fuzzy extractor into a cryptographically strong key, and helper data is generated as a byproduct of the fuzzy extractor's internal ECC method. Finally, the helper data is stored in an external NVM (and hence becomes public information).

(b) **Reconstruction:** the earlier enrolled key is reliably recovered from a noisy *PUF Response* (PR) and the stored helper data as shown in Fig. 2(b). First, the PUF is evaluated again and produces the noisy PR. Next, PR is processed by the fuzzy extractor in combination with the helper data which is retrieved from the external NVM. If the noisy PR is close enough to the PRR obtained during enrollment (i.e. the PUF response is reproducible upto a limited amount of noise), then the extractor succeeds in reliably reconstructing the enrolled key.

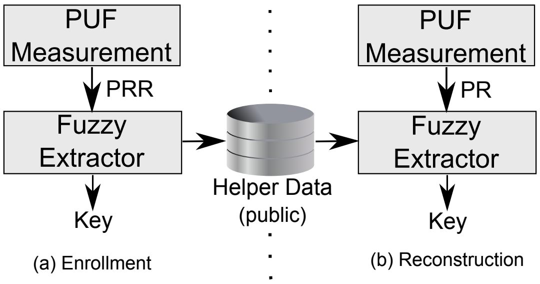

Fig. 2: Operations of a PUF based Key Storage System.

C. PUF Properties

The two most basic quality measures of a PUF implementation are *reproducibility*: expressing how reliable a response can be reproduced on a single device, and *uniqueness*: expressing the difference between responses coming from distinct devices.

1) Reproducibility: A fuzzy extractor needs to be designed to cope with the worst-case expected difference between PRR at enrollment and PR at reconstruction in order to obtain a reliable key generation. The noise on a PUF response is typically expressed as the relative number of bit flips between the enrollment PRR and the PR during reconstruction. The smaller the expected noise, and hence the higher the *reproducibility* of the PUF responses, the more efficient the overall PUF-based key generation system can be implemented.

2) Uniqueness: To generate a secure key, a fuzzy extractor requires that a PUF response is unpredictable, even when other responses on the same PUF or access to other PUFs are given. This entails that:

- The probability that two different PUFs have responses close to each other should be negligible, i.e., PUF responses are highly *unique* and the expected amount of differing bits is close to 50%.
- The bits in a specific PUF response should be highly random and independent, i.e., each bit provides a negligible amount of information about the remaining response bits, and the relative entropy of each response is large.

III. Simulations

A memory system comprising a cell and peripheral circuitry is synthesized and simulated using SPICE, to analyze the reproducibility of memory-based PUFs by adapting the voltage

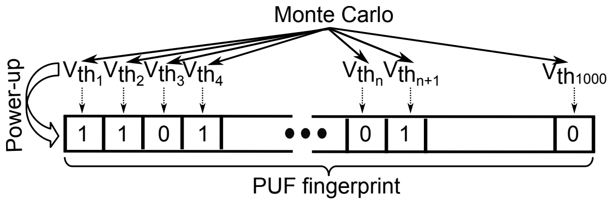
Monte Carlo
Power-up
V_{th_1} V_{th_2} V_{th_3} V_{th_4}
V_{th_n} $V_{th_{n+1}}$
$V_{th_{1000}}$
1 1 0 1 ••• 0 1 0
PUF fingerprint

Fig. 3: SRAM PUF simulation.

ramp-up time to the environmental temperature. In this section, first, the PUF fingerprint generation is presented. Second, the metric used to evaluate noise is discussed. Finally, simulation experiments and results are described.

A. SRAM PUF Response Simulation

Each bit of an SRAM PUF response is generated by an individual SRAM cell. Fig. 3 shows the SRAM fingerprint generation schematic used in our simulations. It has been shown in [17] that the threshold voltage V_{th} of NMOS transistors is the technology parameter with the most impact on the start-up value of an SRAM cell. Hence, Monte Carlo method is used to generate 1000 random values of V_{th} for Q_1 (see Fig. 1) according to the distribution presented in [23], i.e., mean μ = standard NMOS V_{th} and deviation σ = $9\% \cdot \mu$. These 1000 SRAM cells combined create an SRAM cell array that generates a unique and random 1000-bit response after power-up.

B. Noise Metric

To analyze the noise we read the PR of the simulated SRAM cell array for different voltage ramp-up times (t_{ramp}) and different temperatures (*Temp*). Then, the *Fractional Hamming Distance* (FHD) of each measured response compared to the enrollment response (PRR) is calculated; this is the number of differing bits normalized to the response length.

C. Simulation Experiments

To investigate the impact of the voltage ramp-up time t_{ramp} on the noise at different temperatures *Temp*, we consider a range of values for both t_{ramp} and *Temp* for 45nm *Low Power* (LP) [24]. For each combination of *Temp* and t_{ramp} we simulated the power-up of the SRAM cell array 20 times and read its response. The transient noise during power-up is randomly generated by the simulation tool, hence three variable parameters are used for the simulation:

- **Voltage ramp-up time:** $3 \times t_{ramp}$ ($10\mu s$, $50\mu s$ and $90\mu s$),
- **Temperature:** $3 \times$ *Temp* ($-40°$C, $+25°$C, $+85°$C) and,
- **Measurements:** $20 \times$ *Meas* (each with a random seed).

Hence, a total of $(3 \times t_{ramp}) \times (3 \times Temp) \times (20 \times Meas) \times (1000 \times V_{th}) = 180{,}000$ simulations are performed.

D. Simulation Results

Fig. 4 shows the results of FHD calculations per t_{ramp} and $Temp$ considering enrollment performed at $+25°$C with t_{ramp} of (a) 10μs, (b) 50μs and (c) 90μs.

From Fig. 4(a) it can be seen that for *Temp* below the enrollment ($+25°$C), $\max$FHD is lower if t_{ramp} is longer than the one used for enrollment. However, at *Temp* above the enrollment, the opposite is true, e.g., at $+85°$C, key-reconstruction with $10\mu s$ generates the lowest $\max$FHD while at $-40°$C, that is true for $90\mu s$.

Fig. 4(b) and (c) report similar results but now for other t_{ramp} at enrollment ($50\mu s$ and $90\mu s$). Following the trend observed previously, for *Temp* below enrollment ($+25°$C), $\max$FHD is lower if t_{ramp} is longer than the one used during enrollment; e.g., considering Fig. 4(b), at $+85°$C, key-reconstruction with $10\mu s$ generates the lowest $\max$FHD while at $-40°$C, that is true for $90\mu s$.

IV. Discussion

SPICE simulations show that using long t_{ramp} at low temperatures and short t_{ramp} at high temperatures results in reduced SRAM PUF response noise when compared to enrollment. Hence, choosing appropriate t_{ramp} according to ambient temperature, including enrollment, can be used as an efficient scheme to reduce noise and increase reproducibility.

V. Implementation Considerations

The proposed scheme can be implemented by a simple circuit consisting of a voltage regulator and a temperature sensor. Fig. 5 shows an example of such a circuit, comprising five blocks, an SRAM PUF, a voltage ramp-up regulator, an embedded temperature sensor, an ADC and a controller.

The circuit performs five main steps. First, the temperature sensor senses the ambient temperature. Second, this temperature is used as an input to the ADC that converts the given temperature to the closest digital temperature *Temp*. Third, according to *Temp* the Controller is calibrated and the t_{ramp} that minimizes the FHD (noise) is produced. Fourth, the voltage ramp-up regulator powers-up the SRAM PUF with the assigned t_{ramp} and finally, the SRAM PUF generates a PUF response.

One of the main advantages of the proposed optimization technique, besides its evident effectiveness, is that its implementation demands no adaptations of the memory-based PUF circuit itself. In fact the basic PUF comprises only standard library memory cells, but needs to be placed in its own power domain and extended with an embedded temperature sensor and a voltage ramp-up regulator. A small controller regulates the optimal ramp-up time of the memory-based PUF to the sensed temperature, based on a prior calibration. The general design of these extensions is schematically shown for an SRAM PUF in Fig. 5. Since the concerned building blocks are all rather standard, the implementation effort of the proposed optimization technique is considered minimal.

VI. Conclusion

In this paper, we proposed a method based on adapting the voltage ramp-up time to the ambient temperature for enhancing the reproducibility of memory-based PUFs. The combined effect on PUF reproducibility has been evaluated using circuit simulation in 45nm LP CMOS.

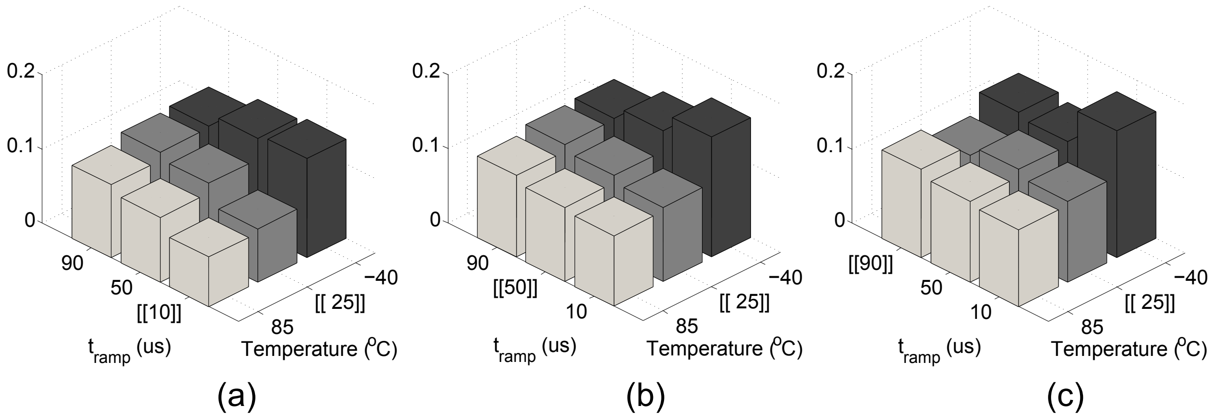

Fig. 4: Maximum Fractional Hamming Distance ($\max$ FHD); enrollment performed at $+25°$C with t_{ramp} of (a) 10μs, (b) 50μs and (c) 90μs.

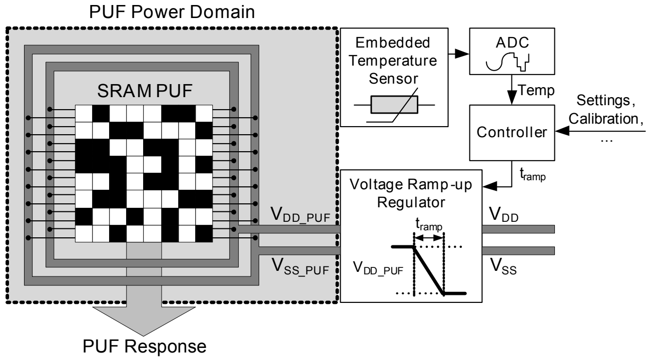

Fig. 5: Schematic of an extended SRAM PUF design.

The results are effective, showing a decrease in PUF noise at extreme temperatures. A significant advantage of the proposed noise-reduction technique is that it can be implemented without altering existing memory-based PUF circuits, but merely by extending them with some relatively standard building blocks. The application of the proposed techniques will result in a significantly reduced complexity and a smaller footprint of a PUF-based key generator. The reproducibility enhancement is achieved while either maintaining or increasing the uniqueness. Future work will include investigating the proposed techniques for alternative memory-based PUFs and other silicon technologies, and implementing the extensions to enable them in silicon.

References

[1] B. Gassend, D. Clarke, M. van Dijk and S. Devadas, "Silicon physical random functions", *ACM CCS*, pp. 148–160, 2002.

[2] R. Maes and I. Verbauwhede, "Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions", *Towards Hardware-Intrinsic Security*, A.-R. Sadeghi and D. Naccache (Eds.), pp. 3–37, 2010.

[3] S. Katzenbeisser, Ü. Kocabas, V. Rozic, A.-R. Sadeghi, I. Verbauwhede and C. Wachsmann, "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon", *CHES*, pp. 283–301, 2012.

[4] A. Maiti, J. Casarona, L. McHale and P. Schaumont, "A large scale characterization of RO-PUF", *HOST*, pp. 94–99, 2010.

[5] T. Yoshida, T. Katashita and A. Satoh, "Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs", *ReConFig*, pp. 298–303, 2010.

[6] J. Guajardo, S.S. Kumar, G.-J. Schrijen and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection", *CHES*, pp. 63–80, 2007.

[7] B. Skoric, P. Tuyls and W. Ophey, "Robust key extraction from Physical Unclonable Functions", *ACNS*, pg. 99–135, 2005.

[8] Y. Dodis, L. Reyzin and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data", *SIAM Journal on Computing*, pp. 97–139, 2008.

[9] J.-P. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates", *AVBPA*, pp. 393–402, 2003.

[10] R. Maes, A. Van Herrewege and I. Verbauwhede, "PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator", *CHES*, pp. 302–319, 2012.

[11] V. van der Leest, B. Preneel and E. van der Sluis, "Soft Decision Error Correction for Compact Memory-Based PUFs Using a Single Enrollment", *CHES*, pp. 268-282, 2012.

[12] M. Hofer and C. Boehm, "An Alternative to Error Correction for SRAM-Like PUFs", *CHES*, pp. 335–350, 2010.

[13] V. Vivekraja and L. Nazhandali, "Circuit-level techniques for reliable Physically Uncloneable Functions", *HOST*, pp. 30–35, 2009.

[14] D. Forte and A. Srivastava, "On improving the uniqueness of silicon-based physically unclonable functions via Optical Proximity Correction", *DAC*, pp. 96–105, 2012.

[15] M. Bhargava, C. Cakir and K. Mai, "Attack Resistant Sense Amplifier based PUFs (SA-PUF) with Deterministic and Controllable Reliability of PUF Responses", *HOST*, pp. 106–111,2010.

[16] R. Kumar, H.K. Chandrikakutty and S. Kundu, "On improving reliability of delay based Physically Unclonable Functions under temperature variations", *HOST*, pp. 142–147, 2011.

[17] D.E. Holcomb, W.P. Burleson and K. Fu, "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Number", *IEEE Transactions on Computers*, vol. 58, no. 9, 2009.

[18] M. Claes, V. van der Leest and A. Braeken, "Comparison of SRAM and FF PUF in 65nm technology", *NordSec*, pp. 47–64, 2011.

[19] S.S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen and P. Tuyls, "The Butterfly PUF protecting IP on every FPGA", *HOST*, pp. 67–70, 2008.

[20] R. Maes, P. Tuyls and I. Verbauwhede, "Intrinsic PUFs from Flip-Flops on Reconfigurable Devices", *WISSec*, 2008.

[21] P. Simons, V. van der Leest and E. van der Sluis, "Buskeeper PUFs, a promising alternative to D Flip-Flop PUFs", *HOST*, pp. 7–12, 2012.

[22] Y. Su, J. Holleman and B. Otis, "A 1.6pJ/bit 96% Stable Chip-ID Generating Circuit using Process Variations", *ISSCC*, pp. 406–611, 2007.

[23] W. Zhao, F. Liu, K. Agarwal, D. Acharyya, S.R. Nassif, K.J. Nowka and Y. Cao, "Rigorous extraction of process variations for 65nm CMOS design", *ESSDERC*, pp. 89–92, 2007.

[24] "http://ptm.asu.edu/", 2012.