Intelligent Voltage Ramp-Up Time Adaptation for Temperature Noise Reduction on Memory-Based PUF Systems

Mafalda Cortez, *Student Member, IEEE*, Said Hamdioui, *Senior Member, IEEE*, Ali Kaichouhi, Vincent van der Leest, Roel Maes, and Geert-Jan Schrijen

***Abstract*—The efficiency and cost of silicon physically unclonable function (PUF)-based applications, and in particular key generators, are heavily impacted by the level of reproducibility of the bare PUF responses (PRs) under varying operational circumstances. Error-correcting codes (ECCs) 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, and hence, an overall more efficient implementation. This paper proposes novel insight and resulting method for reducing noise on memory-based PRs, based on adapting supply voltage ramp-up time to ambient temperature. Circuit simulations on 45 nm low-power CMOS, as well as silicon measurements are presented to validate the proposed method. 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 PR noise up to 3× smaller. In addition, this paper investigates the competitiveness of integrating the proposed method in a commercial product; the investigation is done in two phases. First by determining the saved area, and second by implementing a circuit that maps the ambient temperature into an appropriate voltage ramp-up. The results show that the new system costs up to 82.1% less area while it delivers up to 3× higher reproducibility.**

***Index Terms*—Adapter circuit, memory-based physically unclonable function (PUF), noise reduction, voltage ramp-up time.**

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 [2] for an overview), and their interesting properties are being extensively investigated in large scale experiments [3]–[5]. A silicon PUFs 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 nonvolatile 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: 1) increasing the reproducibility of a typically noisy PUF evaluation to near-perfect reliability and 2) accumulating sufficient unpredictability of possibly low-entropic PUF responses (PRs) 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 (FE) [8], [9] uses error-correcting codes (ECC) to correct noisy PRs. These ECC techniques are very effective in boosting the reliability but tend to be computationally intensive. Moreover, the helper data, which is an unavoidable FE byproduct, will partially disclose the unpredictability of the bare PRs. This needs to be compensated for by using more PUF material and hence, a larger PUF. Both complexity of the ECC decoder and the amount of randomness loss due to the helper data scale with the required error correction capability (ECCap) of the ECC; i.e., less reliable PRs 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, such as temperature and supply voltage variations.

Substantial research effort has been put into reliability enhancement of PUF-based key generators. Careful selection

Manuscript received July 5, 2014; revised November 5, 2014; accepted February 16, 2015. Date of publication April 14, 2015; date of current version June 16, 2015. This work was supported in part by the European Commission through the FP7 Programme under Contract 284833 PUFFIN, and in part by the Dutch "Point One Program" under the RATE Project PNU09C09. This paper was recommended by Associate Editor R. Karri.

M. Cortez and S. Hamdioui are with the Computer Engineering Group, Delft University of Technology, Delft 2628CD, The Netherlands (e-mail: a.m.m.o.cortez@tudelft.nl).

A. Kaichouhi is with the Circuits and Systems Group, Delft University of Technology, Delft 2628CD, The Netherlands.

V. van der Leest, R. Maes, and G.-J. Schrijen are with Intrinsic-ID B.V., Eindhoven 5656AE, The Netherlands.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2015.2422844

of the right ECC algorithms to minimize the helper data loss and decoder implementation cost have been reported [10], [11]. On a physical level, construction improvements to directly decrease the bare silicon PRs noise level have been proposed, either by modifying the PUF circuit [12], [13], or the wafer mask set [14]. Analyzing a silicon PUFs susceptibility to its operating conditions has been explored for reliability enhancement [15], [16].

In this paper, an extension of this paper presented in [26], 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 memory-based PRs. 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 static random-access memory (SRAM) [17]–[19]. This paper shows that intelligent matching of voltage ramp-up time to ambient temperature significantly improves the reproducibility of PRs 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, both using circuit simulation and actual silicon measurements for SRAM PUFs, and only silicon measurements for other memory-based PUF types such as [20]–[23].

In addition, we investigate the competitiveness of integrating the proposed technique in a commercial product. The competitiveness is evaluated first, by investigating the relation between memory-based PUF noise and area overhead, determining the saved area for various technology nodes for various PUF-technologies. Second, by proposing and implementing a circuit that maps the ambient temperature into an adequate voltage ramp-up that minimizes the noise. Comparing the saved area against the area of the circuit that enables the noise reduction, we demonstrate that adapting the voltage ramp-up time to the ambient temperature is a very powerful and industrially attractive technique for memory-based PUFs.

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 discusses the simulation setup, including the noise metric and the simulation results. Section IV details the silicon measurement setup, including the optimization algorithms used, the achieved improvements and their discussion. Section V reviews the various FE constructions, makes the link between area overhead and noise, describes the setup to analyze the impact of noise reduction on the area overhead and presents the results. Section VI provides the requirements, the implementation details, and the results of the circuit that maps the temperature to the voltage ramp-up time. Section VII evaluates the proposed system competitiveness by combining and discussing the previous sections results. Finally, Section VIII concludes this paper.

II. Background: PUFs and Key Generation

This section first briefly provides some preliminaries on the basic operation of memory-based PUFs. Then, it shows how

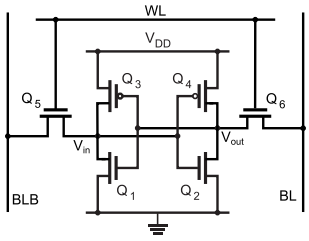

Fig. 1. SRAM cell transistor level schematic.

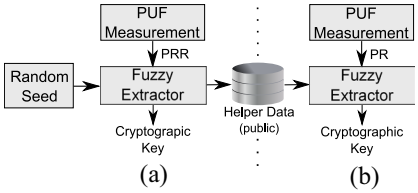

Fig. 2. Operations of a PUF-based key storage system. (a) Enrollment. (b) Reconstruction.

PUFs are deployed in a key storage system, and thereafter it gives the PUFs main quality metrics.

A. Memory-Based PUFs

Memory-based PUFs [6], [20]–[23] 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 peripheral circuitry used to access the cell is comprised by two pass transistors (Q_5 and Q_6), the bitline, complement bitline, and 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 will pull-down its output, forcing the other inverter to pull-up and causing the SRAM cell to settle 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. Hence, an SRAM power-up state is a PR, and this 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 an FE [8], [9], which typically consists of two phases.

1) *Enrollment:* A cryptographic key is generated from a PUF. First, a PUF measurement is taken and used as PUF reference response (PRR). Next, PRR and an external Random Seed are processed by the FE into a cryptographically strong cryptographic key, and helper data is generated as an FE byproduct. Finally, the helper data is stored in an external NVM; hence, it becomes public information.
2) *Reconstruction:* The earlier enrolled cryptographic key is reliably recovered. First, a PUF measurement is taken and used as PR. Typically, some bits of PR are different from the original PRR; hence, PR is a noisy version of PRR. Next, FE processes PR in combination with the

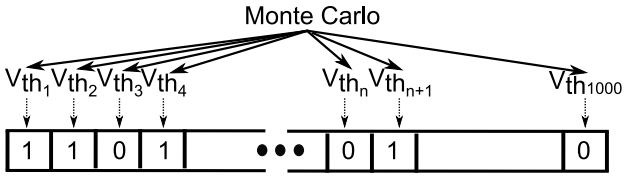

Fig. 3. SRAM PUF simulation.

helper data (retrieved from the external NVM). If PR is close enough to PRR (i.e., PRR is reproducible up to a limited noise amount), then the FE succeeds in reliably reconstructing the enrolled cryptographic key.

C. PUF Properties

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

1) Reproducibility: A FE needs to be designed to cope with the worst-case expected difference between enrollment PRR and reconstruction PR, to reliably generate a key. PR noise is typically expressed as the relative number of bit-flips between the enrollment PRR and the reconstruction PR. The smaller the expected noise, and hence, the higher the reproducibility of the PRs, the more efficient the overall PUF-based key generation system can be implemented.

2) Uniqueness: To generate a secure key, an FE requires PR unpredictability, even if other responses on the same PUF or access to other PUFs are given. This entails the following.

a) The probability that two different PUFs have responses close to each other should be negligible, i.e., PRs are highly unique and the expected amount of differing bits is close to 50%.
b) The bits in a specific PR 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

To analyze the reproducibility of memory-based PUFs when adapting the voltage ramp-up time to the environmental temperature, a memory system comprising a cell and peripheral circuitry is synthesized and simulated using SPICE. In this section, first, the PUF fingerprint generation is presented. Second, the metric used to evaluate noise is discussed. Third, simulation experiments are described. Finally, results are presented and discussed.

A. SRAM PUF Response

Each bit of an SRAM PR is generated by an individual SRAM cell. Fig. 3 shows the SRAM fingerprint generation schematic used in our simulations. Holcomb *et al.* [17] and Cortez *et al.* [18] showed 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, the Monte Carlo method is used to generate 1K random values of V_{th} for Q_1 (see Fig. 1) according to the distribution presented in [24], i.e., mean μ = standard nMOS V_{th} and deviation $\sigma = 9\% \cdot \mu$. These 1K SRAM cells combined create an SRAM cell array that generates a unique and random 1K-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) [17] 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 45 nm low power (LP) [25]. For each combination of Temp and t_{ramp} we simulate 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.

1) *Voltage Ramp-Up Time:* $4 \times t_{\text{ramp}}$ (10, 50, 90, and 130 μs).
2) *Temperature:* $3 \times$ Temp (-40, 25, and 85 °C).
3) *Measurements:* $20 \times$ Meas (each with a random seed).

Hence, a total of $(4 \times t_{\text{ramp}}) \times (3 \times \text{Temp}) \times (20 \times \text{Meas}) \times (1000 \times \text{V}_{\text{th}}) = 240\,000$ simulations are performed.

D. Simulation Results and Analysis

Fig. 4 shows the results of maximum FHD (max FHD) calculations per t_{ramp} and Temp. PUF-based systems are designed to correct up to the worse reconstruction conditions. For this reason, we present the worse (highest) FHD out of the 20 measurements for each of the evaluated conditions. Note that, enrollment is performed at 25 °C with t_{ramp} of Fig. 4(a)–(d) is 10, 50, 90, and 130 μs, respectively; the enrollment conditions are given between "[]" in the figure. From Fig. 4(a), it can be seen that for Temp below the enrollment, 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 μs generates the lowest max FHD while at -40 °C, that is true for 90 μs. Fig. 4(b)–(d) report similar results but now for enrollment at 50, 90, and 130 μs. Following the trend observed previously, for Temp below enrollment, max FHD is lower if t_{ramp} is longer than the one used during enrollment; e.g., Fig. 4(b) shows that the lowest max FHD at 85 °C is achieved with 10 μs while at -40 °C this is realized with 90 μs.

The simulation results revealed a negative correlation between the temperature and the voltage ramp-up time with respect to noise during key reconstruction on memory-PUF fingerprints. The main components of memory-PUFs are MOSFETs; these are vulnerable to three main types of noise: 1) thermal noise (ThN); 2) flicker noise (FN); and 3) shot noise (SN) [34], [35]. During the enrollment phase, we are in fact establishing a noise level reference, that is

$$\text{TN} = \text{ThN} + \text{FN} + \text{SN} \tag{1}$$

where TN is the total noise. First, ThN is related to the scattering of carrier charges in thermal motion, and is directly

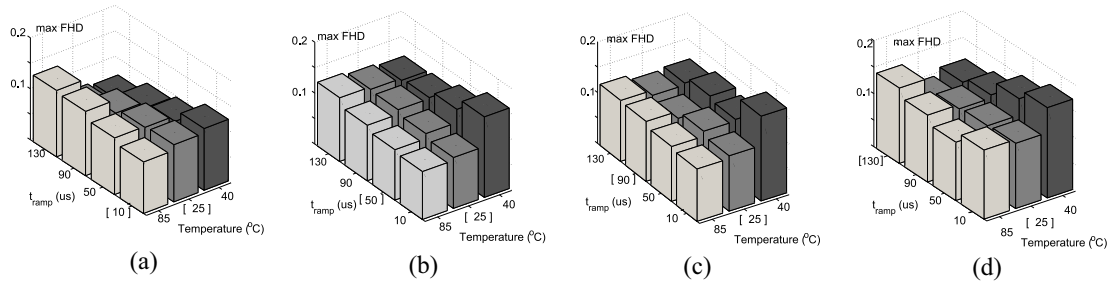

Fig. 4. max FHD; enrollment performed at 25 °C with t_{ramp} of (a) 10 μs, (b) 50 μs, (c) 90 μs, and (d) 130 μs.

proportional to the temperature, i.e., the higher the temperature, the higher the noise. In addition, in short-channel devices, ThN increases with increase in gate-to-source and drain-to-source voltages [34], [35]. Second, FN, also known as $1/f$ noise, is related to trapping and releasing charges near the Si–SiO_2 interface (silicon–silicon dioxide), and is inversely proportional to the frequency. For short-channel devices, a special case of FN occurs—the random telegraph noise (RTN). In fact, FN is the sum of a large amount of RTN [34], [35]. The fingerprints of memory-based PUFs are determined during one single power-up with a certain t_{ramp}; such t_{ramp} can be seen as a part of a periodic signal (e.g., sawtooth signal), and therefore different t_{ramp} corresponds to different signal frequencies influencing FN in different ways. Finally, SN is related to charges overcoming potential barriers, such as moving from the source to the channel; this type of noise is directly proportional to the electrical current [34], [35]. It is worth noting that ThN and FN have much larger impact than SN in the frequency range considered [36]. At higher temperatures, the PUF suffers from higher ThN as compared with enrollment done at lower temperature. To compensate for such noise and get the overall noise close to that of the enrollment, we can reduce the FN at higher temperature reducing the t_{ramp}. At lower temperature, the impact is opposite.

IV. Silicon Validation

To validate the simulation results, we performed silicon measurements on three different types of memory-based PUFs: the SRAM PUF [6], [17], the D flip-flop (DFF) PUF [21], and the buskeeper (BK) PUF [22].

A. Test Setup

The considered memory-based PUF types are manufactured in three different technology nodes. Table I provides an overview of all devices, including the technology node, the number of available integrated circuits (ICs), the number of PUF instances per IC in the given technology (if any), and the total number of tested instances of each PUF type. Note that, each IC contains one or more PUF instances.

Measurements are performed at three different temperatures (−40, 25, and 85 °C) and for ten different t_{ramp} varying from 10 μs up to 500 ms. In case of 40 nm SRAM, the shortest possible t_{ramp} is 50 μs due to specific capacitive load. The measurements flow is as follows.

1) The ICs are placed in a climate chamber and connected to a programmable power supply.
2) Climate chamber is set to one of the test temperatures.

TABLE I
Description of Devices Used in Validation

Technology	# ICs	# PUF inst. / IC			Total # PUF inst.		
		BK	DFF	SRAM	BK	DFF	SRAM
40nm LP	5	-	-	3	-	-	15
65nm LP	50	2	4	4	100	200	200
130nm LP	16	-	1	1	-	16	16

3) ICs are powered with a t_{ramp} from the test set.
4) Each PUF device response is read and stored in a file.
5) The ICs are powered down for 1 s.
6) Steps 3–5 are repeated nine times (i.e., ten measurements per PUF per temperature per t_{ramp}).
7) Change t_{ramp} and repeat steps 3–6 (until all values of t_{ramp} have been tested for this temperature).
8) Change temperature and repeat steps 3–7.

B. Evaluation Metrics

1) Reproducibility: To calculate FHD, first an enrollment response of each PUF instance is measured. Thereafter, each reconstruction measurement is compared to this enrollment by counting the number of flipped bits and dividing it by the response length. A key based on the PR (as described in Section II) is reliable if the worst-case FHD under any stress condition is below the ECCap of the ECC. Hence, the smaller FHD, the lower the required error correction.

2) Uniqueness: We evaluate the uniqueness at enrollment of the different PUF implementations by using: 1) the average between-class Hamming distance (μ-BCHD) [17] and 2) the estimated min-entropy (H_∞) [17] of the measured responses. Note that, for key storage application (as described in Section II) only the uniqueness of the enrollment PR is critical, as it is from this response that the cryptographic key is derived. μ-BCHD is calculated as follows.

1) The enrollment response of each PUF is measured.
2) The Hamming distance between each pair of enrollment responses (i.e., between-class) coming from different PUF instances of the same type is determined (e.g., between all pairs of enrollment responses of 65 nm LP SRAM PUFs are computed).
3) The distribution of these between-class distances is determined and the obtained mean value, normalized to the response length, is μ-BCHD.

Optimally, the obtained distribution should be approximately Gaussian and μ-BCHD should be very close to 50% [17].

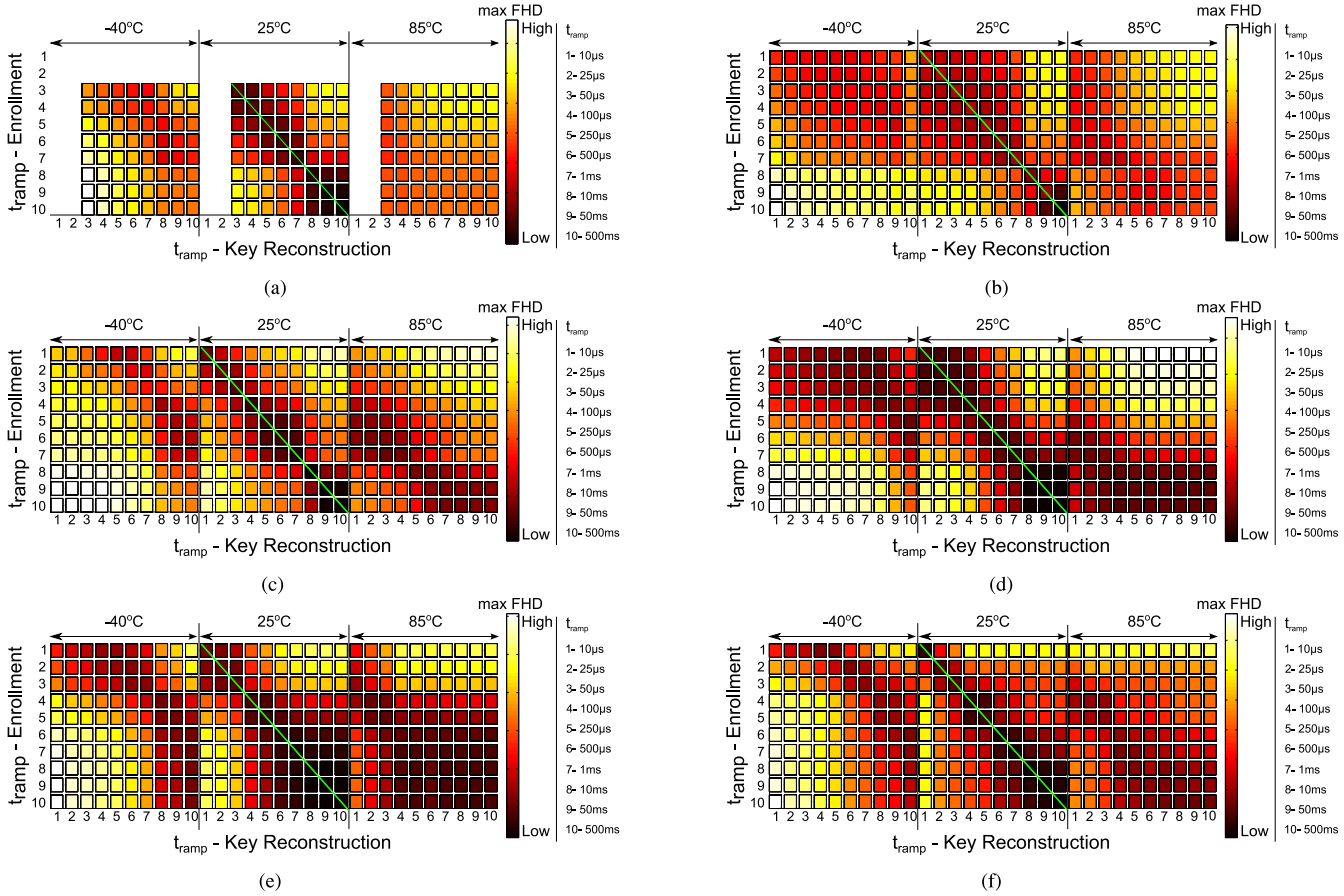

Fig. 5. max FHD for various t_{ramp} enrollment (green line) and key reconstruction. (a) 40 nm, (b) 65 nm, and (e) 130 nm SRAM PUF. (c) 65 nm and (f) 130 nm DFF PUF. (d) 65 nm BK PUF.

H_∞ is used to evaluate the intrinsic unpredictability of PRs. H_∞ is a pessimistic measure of a random variable unpredictability [8]. We estimate H_∞ of the responses of a particular PUF type by considering the following model: each PR bit is assumed to be independent of the other bits in the same response, and that it has an individual probability p_1 of being 1 for a random PUF instance. This model is particularly reasonable for memory-based PUFs, as each response bit originates from an individual memory cell. Under the assumption of this model, $H_\infty = -\log_2 \max\{p_1, 1-p_1\}$ for a single response bit [6]. The value for p_1 of a bit is estimated by counting the number of enrollment responses for which this bit is 1 and dividing by the total number of enrollment responses. The entire response H_∞ is simply the summation of H_∞ of each bit. We express H_∞ as the average H_∞ per bit in a response value, by dividing the total H_∞ of the response by its length. Optimally, H_∞ of a PR bit should be close to 1. Note that, due to the limited number of measured PUF instances, the obtained estimations of H_∞ could be lower than the actual PRs H_∞.

C. Optimization Algorithms

The silicon measurements have the objective to investigate the use of t_{ramp} as a technique for increasing memory-based PR reproducibility (noise reduction). As a side effect, the impact on PUF uniqueness is also investigated. For this purpose, two optimization algorithms are used.

1) Reproducibility Optimization: This algorithm identifies the enrollment t_{ramp} that leads to the highest reproducibility (lowest maximum noise).

2) Uniqueness Optimization: This algorithm identifies the enrollment t_{ramp} that provides the highest H_∞. After this first step the values of t_{ramp} at other temperatures are determined, which minimize the noise.

D. Measurement Results

In order to evaluate the performance of the optimization algorithms, first we analyzed the max FHD for all t_{ramp} enrollment key reconstruction combinations. Fig. 5 shows the results; the t_{ramp} used for enrollment (at 25 °C, also highlighted by a green line) and key reconstruction (at −40, 25, and 85 °C) are represented on the y- and x-axis, respectively, whereas the max FHD is represented by color. These values are obtained using t_{ramp} from 10 μs, which is the shortest feasible t_{ramp} for each PUF, except for 40 nm LP SRAM PUF where the shortest feasible t_{ramp} is 50 μs, up to 500 ms. The max FHD (noise) is determined using ten response measurements per PUF per temperature per t_{ramp}.

Fig. 5(a) reveals a clear convergence pattern toward a local minimum max FHD for each temperature/t_{ramp} enrollment combination. The local minimum max FHD is achieved for t_{ramp} longer than that of enrollment for −40 °C, the same as that of enrollment for 25 °C and shorter than that of enrollment for 85 °C. For example, considering enrollment

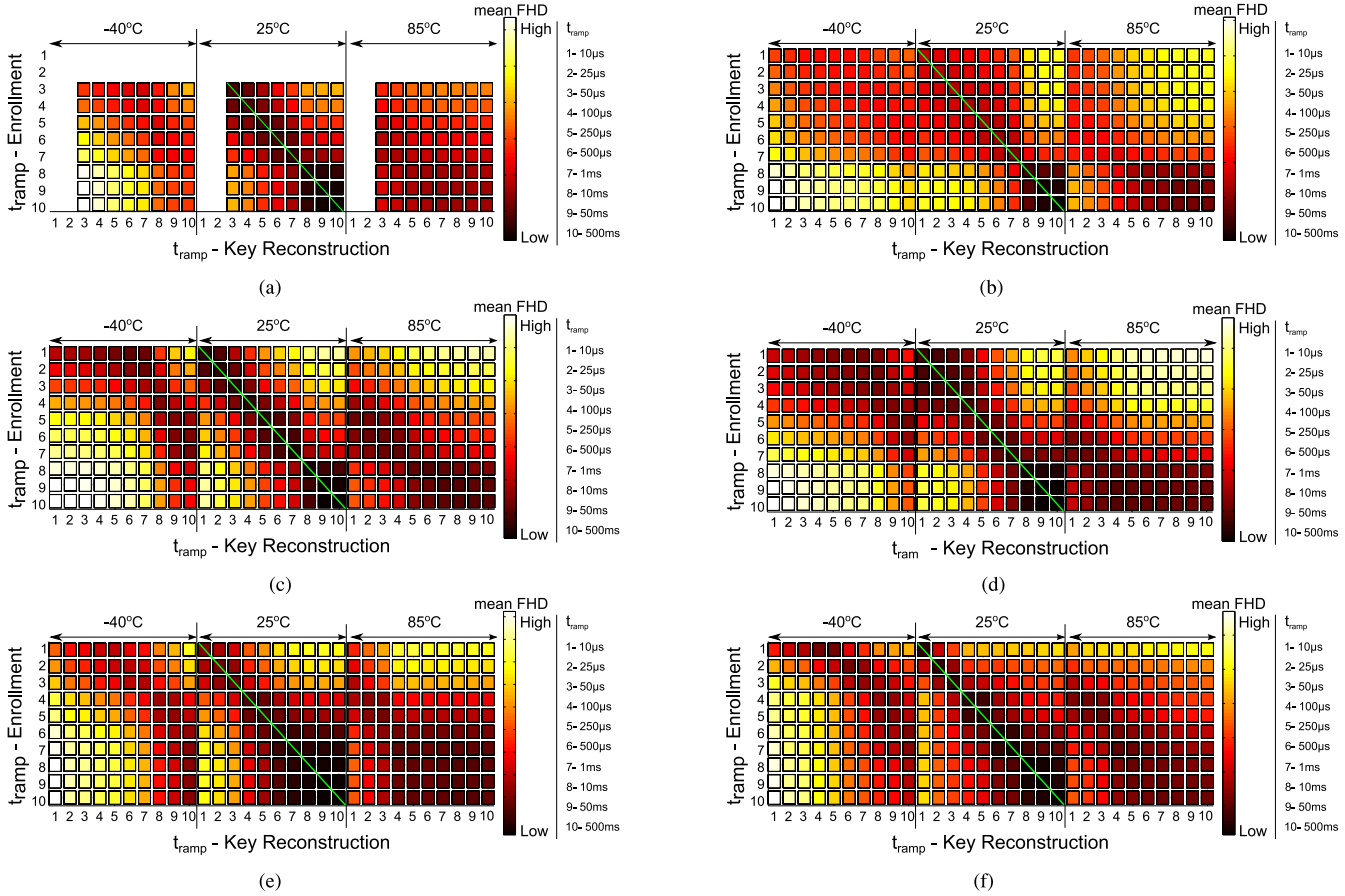

Fig. 6. Mean FHD for various t_{ramp} enrollment (green line) and key reconstruction. (a) 40 nm, (b) 65 nm, and (e) 130 nm SRAM PUF. (c) 65 nm and (f) 130 nm DFF PUF. (d) 65 nm BK PUF.

TABLE II
MEASUREMENT RESULTS WITHOUT OPTIMIZATION

Technology	PUF	t_{ramp}	Maximum noise FHD			μ-BCHD	H_∞
			$-40°$C	$+25°$C	$+85°$C		
40nm LP	SRAM	50μs	23%	6%	20%	0.50	0.73
65nm LP	SRAM	10μs	8%	6%	8%	0.50	0.87
	DFF	10μs	28%	8%	25%	0.37	0.40
	BK	10μs	10.5%	4.5%	20%	0.48	0.75
130nm LP	SRAM	10μs	13%	6%	12%	0.47	0.66
	DFF	10μs	16.5%	5%	28%	0.43	0.61

TABLE III
RESULTS AFTER REPRODUCIBILITY OPTIMIZATION

Technology	PUF	t_{ramp}			Maximum noise FHD			μ-BCHD	H_∞
		$-40°$C	$+25°$C	$+85°$C	$-40°$C	$+25°$C	$+85°$C		
40nm LP	SRAM	10ms	1ms	50μs	14%	4.5%	17%	0.49	0.71
65nm LP	SRAM	50ms	250μs	10μs	7%	5.5%	7%	0.50	0.89
	DFF	50ms	500μs	25μs	11.5%	5%	9%	0.49	0.84
	BK	500ms	1ms	25μs	6.5%	4%	6.5%	0.49	0.81
130nm LP	SRAM	500ms	10ms	1ms	5.5%	2%	4%	0.37	0.42
	DFF	500ms	10ms	500μs	9.0%	3.0%	9.0%	0.46	0.63

3 (i.e., t_{ramp} at 50 μs), for -40 °C max FHD decreases until t_{ramp} 5, increasing thereafter, while for both 25 and 85 °C max FHD increases with t_{ramp} increase. Similar trends are observed for the other PUF types and technology nodes.

Table II summarizes the information of Fig. 5 for the shortest enrollment t_{ramp} per PUF type; i.e., it shows the original measured maximum noise values for the considered temperatures. Moreover, it shows the uniqueness indicators. Table II is used as reference to compare the results of the proposed optimization algorithms against, as the enrollment conditions are the standard ones.

Table II reveals that, overall, the maximum noise measured is 28% at -40 °C (for the 65 nm DFF PUF), 8% at 25 °C (for the 65 nm DFF PUF), and 28% at 85 °C (for the 130 nm DFF PUF). Regarding uniqueness, although a truly fair comparison is not possible due to limited available devices per technology node and PUF type, the 65 nm DFF PUF has the lowest μ-BCHD $= 0.37$ and $H_\infty = 0.40$.

In addition, to investigate whether the observed convergence toward a local minimum holds for the mean FHD, we perform a similar analysis as for max FHD. Fig. 6 shows the results. The mean FHD (noise) is determined using ten response measurements per PUF per temperature per t_{ramp}. Fig. 6 reveals the same convergence trend observed in Fig. 5.

1) Reproducibility Optimization: Table III presents the reproducibility optimization algorithm results; it shows the t_{ramp} configuration that minimizes the noise (maximizes reproducibility) per temperature in comparison to enrollment. The results reveal that for all tested PUFs, adapting t_{ramp} to the ambient temperature has a major impact on the maximum noise. For low temperatures, noise reduction is realized with

TABLE IV
RESULTS AFTER UNIQUENESS OPTIMIZATION

Technology	PUF	t_{ramp}			Maximum noise FHD			μ-BCHD	H_∞
		$-40°$C	$+25°$C	$+85°$C	$-40°$C	$+25°$C	$+85°$C		
40nm LP	SRAM	1ms	100μs	50μs	16%	6%	19%	0.50	0.73
65nm LP	SRAM	50ms	100ms	50μs	13%	2%	8%	0.50	0.89
	DFF	500ms	10ms	250μs	18.5%	2.5%	8%	0.50	0.90
	BK	100ms	250μs	10μs	7%	5%	9%	0.50	0.88
130nm LP	SRAM	1ms	10μs	10μs	7.5%	6%	12%	0.47	0.66
	DFF	50ms	500μs	10μs	10%	4.5%	9.5%	0.47	0.67

longer t_{ramp}; whereas for high temperatures this is realized with shorter t_{ramp}. For example, the maximum noise for 65 nm LP DFF PUF at -40 °C with $t_{\text{ramp}} = 10$ μs for both enrollment and reconstruction is originally 28%. However, if the optimized t_{ramp} is used both at enrollment (500 μs at 25 °C) and at reconstruction (50 ms at -40 °C), then the maximum noise is reduced to merely 11.5%. Note that, all results of Table III follow the same trend, as predicted by the simulation results of Section III-D. Since this algorithm does not optimize uniqueness, μ-BCHD and H_∞ are deteriorated for some PUFs (e.g., 130 nm SRAM PUF), while they are significantly improved for others (e.g., 65 nm DFF PUF).

2) Uniqueness Optimization: Table IV reports the uniqueness optimization algorithm results; it shows: 1) the t_{ramp} at enrollment that maximizes uniqueness and 2) the t_{ramp} for the other temperatures that results in the lowest maximum noise (with respect to the t_{ramp} selected for enrollment). Uniqueness indicators μ-BCHD and H_∞ are at least as high as the originals for 40 and 130 nm SRAMs, and for the remaining devices these indicators are higher than the original indicators. The uniqueness optimization algorithm clearly leads to significant improvements in μ-BCHD and H_∞ for DFF and BK PUFs. However, this improvement is negligible for the SRAM PUFs for all tested nodes. Since this algorithm does not select the enrollment t_{ramp} optimized for reproducibility, it is natural that the noise resulting from this algorithm is worse than that of reproducibility optimization algorithm. In case of the 65 nm SRAM PUF, the maximum noise at -40 °C is even worse than the measurements without optimization. Reason for this is that t_{ramp} at enrollment (25 °C) is very long and the algorithm is unable to find a corresponding longer t_{ramp} at -40 °C.

E. Discussion

SPICE simulations show that using long t_{ramp} at low temperatures and short t_{ramp} at high temperatures results in reduced SRAM PR noise when compared to enrollment. The observation is validated using silicon measurements, and holds for all technology nodes and memory PUF type investigated. Hence, choosing appropriate t_{ramp} according to ambient temperature, including enrollment, can be used as an efficient scheme to reduce noise and increase reproducibility.

Moreover, the silicon measurements have also indicated that varying t_{ramp} can have a significant impact on the uniqueness of memory-based PUFs. We can conclude from our measurements that t_{ramp} can slightly bias the fingerprints of memory-based PUFs. The bias is visible by the uniqueness metrics, as these represent the correlation between fingerprints during enrollment. When selecting a certain t_{ramp} we are either enhancing this bias behavior (for reliability optimization) or neutralizing it (for uniqueness optimization); e.g., a PUF device that would generate a response of only 1s would be 100% reliable (FHD = 0), however, it would not be unique.

By choosing the proper optimization algorithm according to the PUF type, noise can be reduced when compared to the original results in Table II while either maintaining or increasing the uniqueness indicators. Inspecting the silicon results with respect to reproducibility and uniqueness reveals the following.

1) The 40 and 65 nm SRAM PUFs benefit from applying the reproducibility optimization algorithm, but the uniqueness optimization algorithm is not very effective as there is very little margin for improvement. Furthermore, the uniqueness optimization algorithm does not significantly minimize the noise for the tested SRAMs.
2) The 130 nm SRAM PUFs benefit from applying the uniqueness optimization algorithm, as the noise is reduced while the uniqueness is maintained.
3) BK and DFF PUFs benefit from applying the uniqueness optimization algorithm, since the original silicon results show that there is a lot of room for improvement. Besides increasing the PR uniqueness, the proposed algorithm also decreases the noise at -40 and 85 °C temperatures. Hence, this algorithm works very well for these PUF types.

V. NOISE REDUCTION IMPACT ON AREA OVERHEAD

In this section, we investigate the noise reduction impact on the area overhead of memory-based PUFs by means of adapting the voltage ramp-up time to the temperature. First, we briefly describe the FE and its possible configurations. Then, we relate noise with area overhead. Thereafter, we define a set of experiments to investigate the impact noise reduction has on the area overhead. And finally, we show and discuss the results of the experiments.

A. Types of Fuzzy Extractor Constructions

An FE is a fundamental component of a PUF-based key storage system (see Fig. 2); it has two main functions.

1) *Information Reconciliation:* It uses the helper data to correct errors on the measured PR.
2) *Privacy Amplification:* Considering that the helper data contains information on the PRR, privacy amplification is needed to make sure that the helper data does not reveal any information on the derived cryptographic key.

The FE compresses the resulting data into a cryptographic key with maximum entropy making it impossible for an attacker to guess the key [8], [9]; it also removes any biasing (unequal distribution of zeros and ones) in the error-corrected PR.

Information reconciliation is enabled by error correction blocks, while privacy amplification is enabled by hash function, see Fig. 7. The number and type of error correction blocks depends on both noise and application of each PUF-based system. Encoder blocks are used to add redundancy to the original data during the enrollment phase, while decoder blocks aim at recovering the original data during the key reconstruction phase. The hash function concludes this phase.

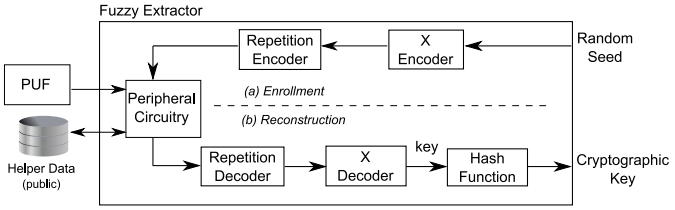

Fig. 7. FE.

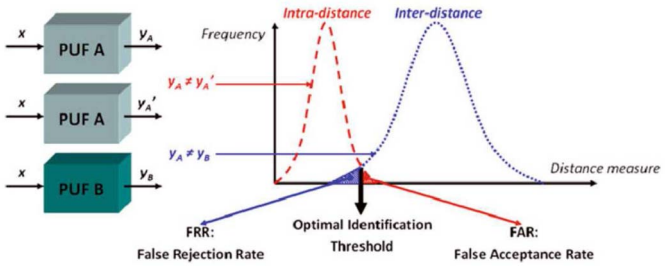

Fig. 8. FRR and FAR [27].

There are several popular constructions with respect to the type of error correction blocks and their parameters. Error correction blocks can be classified into block codes and convolutional codes. Block codes are memoryless, i.e., the encoder's output at any given time depends only on the input at that time. They are easy to implement, efficient with small data, and have low area overhead. However, they suffer from lower ECCap when compared to the convolutional codes. On the other hand, convolutional codes have memory, i.e., the encoder's outputs at any given time (t) depends not only on the inputs at that time unit but also on some of previous inputs. They have higher ECC capabilities. However, convolutional codes require long data streams to work efficiently, are complex to implement and have higher area overhead. For these reasons, block codes are the most used in FE for PUF-based systems.

There are various types of FE constructions using linear block codes for error correction; typical constructions comprise repetition code followed by either Golay code or Reed–Muller code [27]. The aforementioned FE constructions owe their popularity to their area overhead efficiency when compared with their Bose Chaudhuri Hocquenghem counterparts, while delivering the same error correction efficiency [27], [28]. For this reason this paper focus on these FE constructions. Fig. 7 depicts a generic FE; it comprises a repetition code and a generic X code representing either a Golay [24, 12, 8] code, or a Reed–Muller16 [16, 5, 8] code, or a Reed–Muller8 [8, 4, 4] (note that, the used codes have n length, k secret bits, and d minimum Hamming distance, resulting in $[n, k, d]$).

B. Linking Noise Reduction to Area Overhead

A high quality PUF-based system is the one which: 1) efficiently reconstructs a valid cryptographic key from a true PUF device (the one used for enrollment) under various conditions and 2) does not reconstruct a valid cryptographic key from a false PUF device (any device different than the one of enrollment being illegally used to reconstruct the key of the true device). Common quality metrics used for PUF-based systems are false rejection rate (FRR) and false acceptance rate (FAR) [28]. FRR is the probability that the noise of a PR of true device A is above the error correction capabilities of the PUF-system and therefore, the authentication of true device A is rejected. FAR is the probability that the noise of a PR of device B is such that it is mistakenly corrected to the PRR of device A and therefore, device B is falsely authenticated as A. FRR and FAR are exemplified in Fig. 8 [27]; the figure shows two FHD histograms: the intra-FHD (i.e., noise) on the left side and the inter-FHD (i.e., the FHD among different devices) on the right side. When designing a PUF-based system, ideally all intra-FHD would be corrected and at the same time each device would be perfectly distinguishable from others. However, in reality the two histograms overlap, resulting into two different areas FRR and FAR. The optimal identification threshold is when FRR = FAR.

To investigate the impact of PUF noise reduction on the area of PUF system, we need to estimate the quality metrics as a function of the raw PR and ECCap. Let us consider the system shown in Fig. 7. During key reconstruction, an FE is able to successfully reconstruct the cryptographic key only when the output of the X decoder is correct for all decoding iterations (note that, the successful reconstruction tolerates errors at the output of the repetition decoder, as long as these errors are corrected by the X decoder); i.e., when the number of errors are within error correction capabilities of the PUF-system. Assume that the hash function needs an input key with a length "l" to produce the required cryptographic key; the key is generated by multiple iterations of the decoding path (i.e., repetition decoder combined with X decoder). In addition, assume that the number of secret bits per decoding iteration is k [28]; these bits reflect the original information coming from the PUF and the random seed (see Fig. 7), and not the redundant bits introduced by the encoding and decoding. To generate key with a length l, we need $\lceil l/k \rceil$ decoding iterations. The probability that a true key is not reconstructed can be expressed as [28]

$$\mathrm{FRR} = 1 - (1 - \mathrm{PE}_{\mathrm{Xcode}})^{\mathrm{iterations}} \tag{2}$$

where $\mathrm{PE}_{\mathrm{Xcode}}$ is the probability that one or more errors occur above the error correction capabilities of X decoder. Note that, $(1 - \mathrm{PE}_{\mathrm{Xcode}})^{\mathrm{iterations}}$ denotes the probability that all errors are corrected for all the decoding iterations. $\mathrm{PE}_{\mathrm{Xcode}}$ can be expressed as [28]

$$\begin{aligned}\mathrm{PE}_{\mathrm{Xcode}} &= \sum_{i=t+1}^{s} \binom{s}{i} \mathrm{PE}_{\mathrm{rep}}^{i} (1 - \mathrm{PE}_{\mathrm{rep}})^{s-i} \\ &= 1 - \sum_{i=0}^{t} \binom{s}{i} \mathrm{PE}_{\mathrm{rep}}^{i} (1 - \mathrm{PE}_{\mathrm{rep}})^{s-i}\end{aligned} \tag{3}$$

where t and s are X decoder ECCap and code length, respectively, and $\mathrm{PE}_{\mathrm{rep}}$ is the probability that one or more errors occur above the error correction capabilities of repetition decoder (see Fig. 7). $\mathrm{PE}_{\mathrm{rep}}$ can be estimated as [28]

$$\begin{aligned}\mathrm{PE}_{\mathrm{rep}} &= \sum_{i=\lceil n/2 \rceil}^{n} \binom{n}{i} \epsilon^{i} (1 - \epsilon)^{n-i} \\ &= 1 - \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{i} \epsilon^{i} (1 - \epsilon)^{n-i}\end{aligned} \tag{4}$$

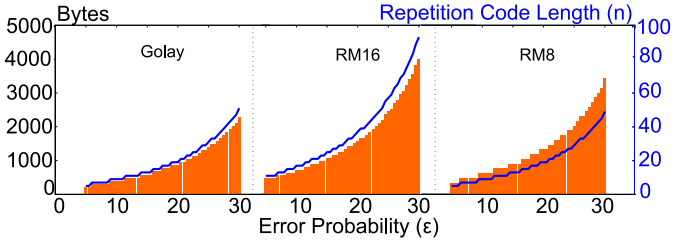

Fig. 9. PUF size (bars) and repetition code length (line) versus error probability (ϵ).

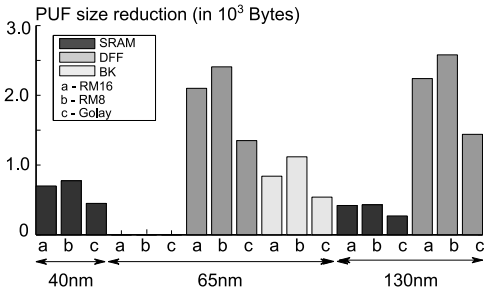

Fig. 10. Absolute PUF size reduction.

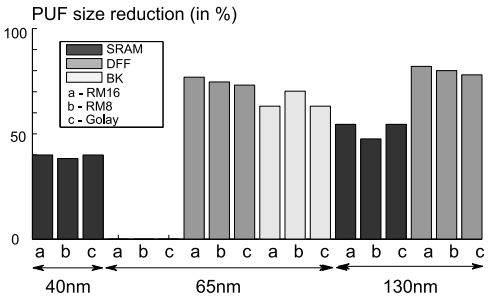

Fig. 11. Relative PUF size reduction.

where n is the repetition decoder code length and ϵ is the PUF-response error probability, see Fig. 7. Note that, the repetition encoder is not used during Key Reconstruction. Using the previous equations, one can easily determine n as a function of ϵ for a given FRR and s; say $n = f(\epsilon)$.

The size of the required PUF data can be estimated [28]

$$\begin{aligned} \text{PUF}_{\text{bits}} &= (\text{code length X decoder}) \\ &\quad \times (\text{code length repetition decoder}) \\ &\quad \times (\text{number iterations}) \\ &= s \times n \times \text{iterations} \\ &= s \times \text{iterations} \times f(\epsilon). \end{aligned} \tag{5}$$

C. Simulation Setup

To estimate the noise reduction impact on PUF-based systems area for several FE construction types, we use the equations introduced in the previous section with the following set of values.

1) FRR $= 10^{-6}$ [28].
2) The key (input of hash function) has a length $l = 171$ bits; here, we assume that we want to generate a key of 128 bits of entropy and we consider a secrecy rate (minimal amount of compression that needs to be applied to a PUF fingerprint by the hash function) of 0.75 [6], [27], hence, $\lceil 128/0.75 \rceil = 171$ bits are required [27].

In addition, we perform the simulation for the following scenarios.

1) Fifty-one different ϵ; we sweep ϵ from 5% up to 30% with a step of 0.5%.
2) Three different combinations of s and k; these reflect three FE constructions: a) Golay-based with $\{s, k\} = \{24, 12\}$; b) RM16-based with $\{s, k\} = \{16, 5\}$; and c) RM8-based $\{s, k\} = \{8, 4\}$.

D. Results and Analysis

Fig. 9 shows the results for each of the three FE constructions investigated; the left y-axis (bars) depicts the required memory (in bytes), the right y-axis (line) the required repetition code length, and the x-axis the PR error probability ϵ. From the figure we can make the following conclusions.

1) Reduction in noise ϵ significantly reduces the required PUF size and n. Regardless of the FE construction, the lower ϵ, the lower the PUF size and the lower the repetition code length. For example, when $\epsilon = 15\%$, a RM16-based PUF system requires 910 PUF bits and a repetition code of length $n = 13$, while when $\epsilon = 5\%$ only 350 PUF bits and $n = 5$ are required to realize the same quality (FRR); hence, a noise reduction of 3× causes a 2.6× reduction in both PUF size and n.
2) Golay-based and RM16-based PUF systems are the ones benefiting the most from our technique; their PUF size and n reduces by 2.6× when ϵ reduces from 15% to 5%. However, this is only 2.3× for RM8-based. Moreover, overall, Golay-based PUF system is the one with smaller PUF size and n for any given ϵ.

Now that we have determined the PUF size as a function of the noise, we can estimate the saved PUF size based on our method by first estimating the PUF size of the PUFs shown in Table II (without voltage ramp-up optimization) and thereafter for those shown in Table III (with voltage ramp-up optimization). This will be done as follows.

1) For each of the PUFs in Table II, select the maximum noise FHD (=ϵ), and use Fig. 9 to calculate the required PUF size.
2) For each of the PUFs in Table III, select the maximum noise FHD, and use Fig. 9 to calculate the required PUF size.
3) Determine the savings in PUF size by subtracting the PUF size values found in 2) from those found in 1).

The results are plotted in Figs. 10 and 11. Fig. 10 shows the absolute PUF size reduction while Fig. 11 shows the relative PUF size reduction. The results show that the area savings are strongly PUF type and FE construction dependent. DFF PUFs are the ones benefiting the most; e.g., 130 nm DFF RM16-based requires 2.24K bytes less of PUF material, i.e., a reduction of 82.1%. On the other hand SRAM PUFs are the ones benefiting the least; although that for 40 and 130 nm a quite saving is achieved for all FE constructions, almost no saving is realized for 65 nm irrespective of the FE construction. This is due to the small improvement that the optimization algorithm has on this PUF type, see Tables II and III.

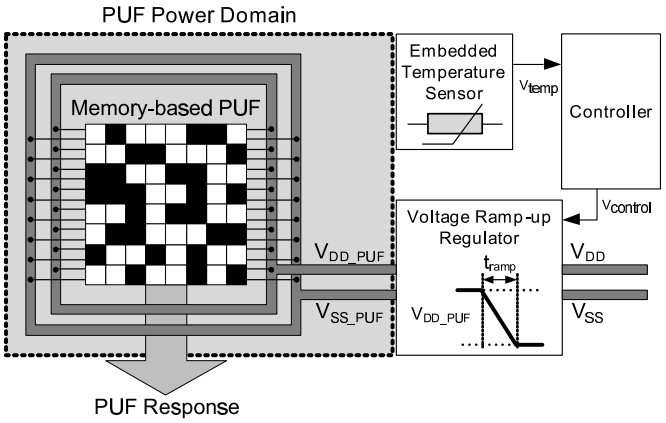

Fig. 12. Schematic of an extended memory-based PUF design.

VI. Adapter-Circuit Implementation

The proposed noise reduction scheme can be implemented by a simple circuit consisting of a temperature sensor, a controller and a voltage regulator. In this section, first we define the requirements of such a circuit. Then, we propose and implement our solution. Finally, we extract the circuit characteristics and discuss them.

A. Requirements

We divide the requirements into design requirements and functional requirements. From design perspective the proposed noise reduction scheme has an added value only if the area of the circuit (which enables various t_{ramp} according to the sensed temperature) is less than the area of the memory it saves. As seen in the previous section, the saved area varies with technology node, memory-PUF type, and FE construction. Due to this, we have different area budgets for the different scenarios, ranging from virtually 0 GE (for 65 nm SRAM PUF) up to 20 kGE (for the 65 nm DFF PUF); gate equivalent (GE) is a technology node independent metric of area that denotes the area of NAND2 with standard drive strengths. Note that, 1 GE is considered as a reasonable estimate of a single SRAM, DFF or BK cell for any of the investigated technologies according to [29]–[31].

In addition, as PUF-based systems are active only during the start-up of a device to generate the key, delay, and power consumption play very minor roles. Therefore, we consider the area overhead to be our main design requirement.

With respect to functional requirements, a set of targets is defined. Table III shows that the optimal t_{ramp} per sensed temperature varies with technology node and memory-PUF type. Hence, as there are several possible configurations, we decided to target the extreme values of t_{ramp}; i.e., t_{ramp} 10 μs at 85 °C, 1 ms at 25 °C, and 500 ms at −40 °C.

In short, the requirements are as follows.

1) Low area overhead (up to budget).
2) Output $t_{\text{ramp}} = 10$ μs at 85 °C, $t_{\text{ramp}} = 1$ ms at 25 °C, and $t_{\text{ramp}} = 500$ ms at −40 °C.

B. Adapter-Circuit

Fig. 12 shows the block diagram of a memory-based PUF extended with the adapter circuit. This system comprises four blocks: a memory-based PUF, an embedded temperature sensor, a controller, and a voltage ramp-up regulator. It performs five main steps. First, the temperature sensor senses the ambient temperature and outputs V_{temp}. Second, V_{temp} is used as the input to the controller, which accordingly, generates a calibration voltage V_{control}. Third, V_{control} is used as an input to the voltage ramp-up regulator, which outputs a t_{ramp} that minimizes the FHD (noise). Finally, the memory-based PUF is powered-up with the assigned t_{ramp}, generating a PR.

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, a voltage ramp-up regulator and controller. The general design of these extensions is schematically shown in Fig. 12. Since the concerned building blocks are all rather standard, the implementation effort of the proposed optimization technique is considered minimal.

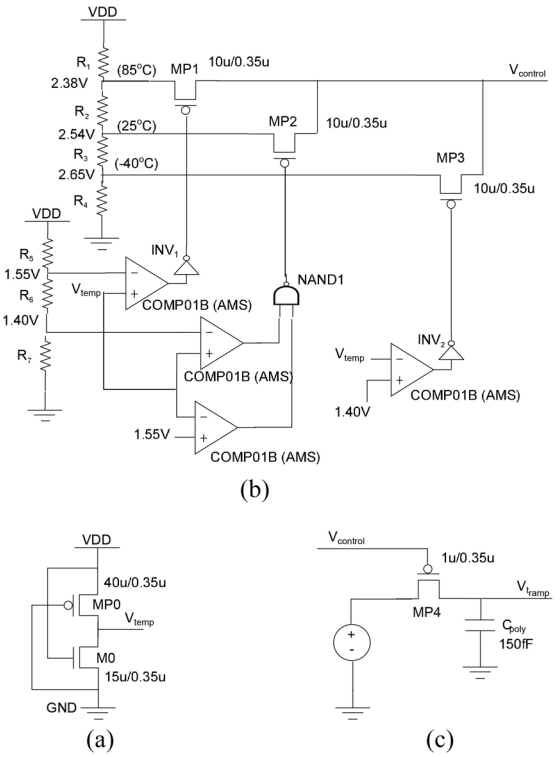

Fig. 13. Adapter circuit schematic. (a) Temperature sensor. (b) Controller. (c) Voltage ramp-up regulator.

C. Implementation

Fig. 13 shows the schematic of the circuit; where Fig. 13(a) depicts the embedded temperature sensor, Fig. 13(b) the controller, and Fig. 13(c) the voltage ramp-up regulator. The circuit is implemented in 0.35 μm, due to lack of availability of smaller technologies, and with AMS technology. We implement a temperature sensor comprising two MOSFETs (MP0 and M0). The sensor outputs a voltage (V_{temp}) that is proportional to the sensed temperature.

The controller, Fig. 13(b), is an intermediary circuitry that maps its input voltage V_{temp} to its output voltage V_{contol}. Each one of the three pMOS (one pMOS per voltage ramp-up time) has at its drain the specific voltage that is required for the voltage ramp-up regulator to deliver the specific t_{ramp}; MP1 for 85 °C, MP2 for −40 °C and MP3 for 25 °C. When a certain temperature is sensed, only the pMOS transistor that represents the closest temperature should drive. The selection of the driving transistor is done via the operational amplifiers, which are used as comparators in this configuration. The voltage outputted by the temperature sensor is compared against the reference values for each temperature. For the extreme temperatures (i.e., −40 and 85 °C) only one comparison is required as we only need to make sure that the V_{temp} is either above (for 85 °C) or below (for −40 °C) the reference voltage of the respective temperature. For intermediary temperatures (i.e., 25 °C) two comparisons are required (hence, two operational amplifiers) as we need to make sure that the received V_{temp} is above a reference and below another. The output of the comparisons for the extreme temperatures needs to be inverted (INV0 and INV1) as pMOS are active for low-voltage at their gates. With the output of the two comparisons of the intermediary temperature we perform an AND (AND0) operation as MP3 should be driven only when both comparisons are true. Finally, two networks of voltage dividers (one comprised by R1, R2, R3, and R4, and the second comprised by R5, R6, and R7) are used to define the reference voltages at the drain of the pMOS and at the inputs of the operational amplifiers, respectively.

The voltage ramp-up regulator, Fig. 13(c), is a basic RC circuit, where the resistor has been replaced by an MOSFET. By varying the voltage at the gate of the MOSFET MP4 we can tune its resistance such that the time constant of the circuit is the one of our specifications (i.e., 10 μs at 85 °C, 1 ms at 25 °C, and 500 ms at −40 °C).

It is worth emphasizing that the proposed circuit generates more than just the three specified voltage ramp-up times for enrollment and extreme temperature corners. The voltage ramp-up time decreases monotonically from 500 ms down to 10 μs, as the temperature increases from −40 °C up to +85 °C; from the continuous range of voltage ramp-up times, we fix the values for the enrollment and extreme corners. The voltage ramp-up times for the remaining temperatures are intrinsically generated by the change in the resistance of the MOSFET MP4 of the voltage ramp-up regulator. This feature is a big plus of the design as it provides larger voltage ramp-up time granularity while not increasing the area overhead of the circuit.

D. Results

The results show that the circuit successfully maps the ambient temperature into the required voltage ramp-up time.

Fig. 14 shows the results for the voltage ramp-up regulator circuit; the circuit outputs at −40 °C a t_{ramp} of 500 ms, at 25 °C a t_{ramp} of 1 ms, and at 85 °C a t_{ramp} of 10 μs, as required. Moreover, as predicted, the voltage ramp-up time decreases continuous and monotonically from 500 ms down to 10 μs, as the temperature increases from −40 °C up to 85 °C; e.g., at −30 °C the circuit outputs a t_{ramp} of 358 ms, while at 75 °C it outputs a t_{ramp} of 12.6 μs. These results reveal the extra resolution of the circuit, which is realized for free (i.e., with no extra area overhead). The voltage ramp-up regulator has an area of 563.36 μm^2 (22.4 × 25.15 μm) which is fixed regardless of the resolution of the system and it is easily implementable in other technology nodes.

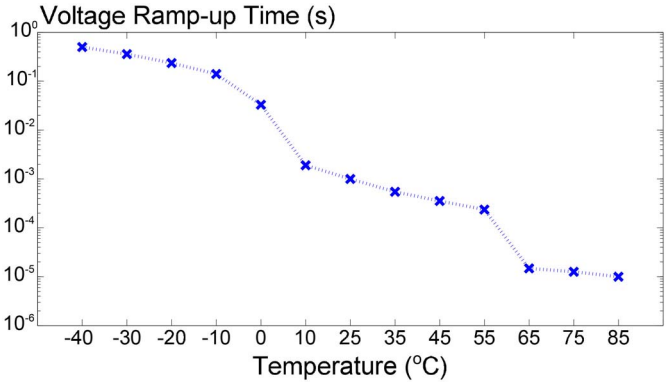

Fig. 14. Voltage ramp-up time versus temperature.

The controller, as designed, outputs one of the three reference voltages (Vm40 = 2.65 V, V25 = 2.53 V or V85 = 2.38 V). It has an area of 0.014 mm^2 (71.5 × 204.5 μm), which 90% corresponds to the area of the operational amplifiers (area of one operational amplifier 0.0034 mm^2). The controller is easily implementable in other technology nodes.

The temperature sensor outputs a voltage with a linear relation with the temperature; V_{temp} is 1.32 V at −40 °C, 1.47 V at 25 °C, and 1.61 V at 85 °C, which results in a resolution of 2.5 mV/°C. The temperature sensor has an area of 169.035 μm^2 (8.85 × 19.1 μm). Moreover, the sensor has a fixed area regardless of the resolution of the system and it is easily implementable in other technology nodes.

Overall, the circuit has an area overhead of 0.015 mm^2 (70.9 × 214.75 μm).

VII. Discussion and Comparison

In this section, first we discuss the impact of our scheme on area overhead, second that of on the delay and finally we discuss the procedure for investigating the temperature/voltage ramp-up time for other PUFs.

A. Impact on Area Overhead

To evaluate the attractiveness of integrating the adaptive circuit when compared with the classic approach, we need to determine the overall area before and after the optimization and compare them. As the adaptive circuit and the investigated memory-PUFs are implemented in different technology nodes, we cannot directly compare the areas; we need a fair comparison metric. Therefore, we convert the area of the adaptive circuit to GE according to [33]; 0.015 mm^2 corresponds to 275 GE (= $\lceil 0.015\ \text{mm}^2/54.6\ \mu m^2 \rceil$, where 54.6 μm^2 corresponds to the area of NAND2 cell in 0.35 nm [33]). We can determine the overall reduction in area overhead as follows. Add the 275 GE of the adaptive circuit to that of the PUF-system after the optimization and compare it with the PUF-system before the optimization. The results are depicted in Figs. 15 and 16. Fig. 15 shows the area overhead,

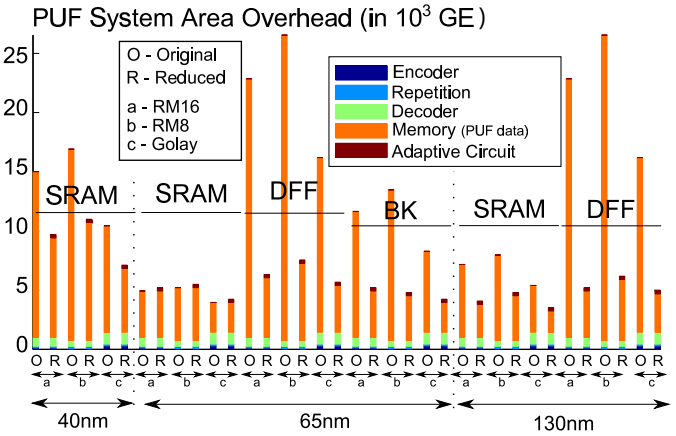

Fig. 15. Absolute area overhead without (O) and with (R) noise reduction.

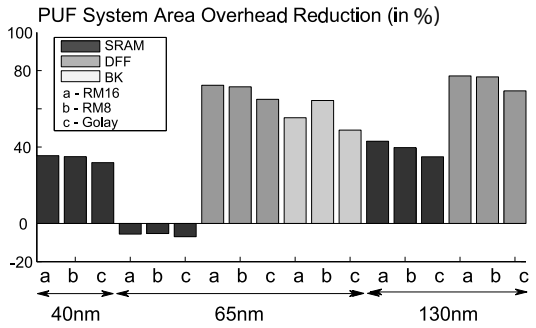

Fig. 16. Relative area overhead reduction.

before (Original) and after (Reduction) the noise reduction, for the different PUF-systems constructions investigated. The area overhead values of encoders, repetition and decoders, for the various constructions, were extracted from [28]. Fig. 16 shows the relative area overhead reduction in percentage. From Fig. 15, we can conclude the following. First, for all memory-PUF system constructions, the block that impacts the most the area overhead is the memory (PUF data size). Therefore, methods targeting noise reduction (resulting in memory reduction), such as the one proposed in this paper, are good allies to reduce the overall cost of the system. Second, the area overhead of Golay, RM16 or RM8 is not impacted by the noise reduction; the implementation of these blocks is independent from PUF noise (ϵ) as these encode/decode a standard number of bits per iteration.

Note that, in the figure we assumed the area overhead of the repetition code as constant. This is a conservative assumption, as in truth, the area overhead of this block is reduced as the noise decreases. As seen in [32], the repetition code hardware implementation comprises a counter, which counts up to n (length of the repetition code). The higher the n the higher the area overhead of the counter, hence, the higher the area overhead of the repetition code. We have seen in Fig. 9 that n decreases with noise, and so decreases the area overhead of the repetition code. Therefore, the overall area reduction is slightly greater than the one presented.

Considering both figures reveals that, overall, integrating the adapter circuit in a memory-based PUF system is an attractive solution. Five out of the six investigated PUF memories have their area overhead reduced, ranging from a minimum of 31.6% (40 nm SRAM) up to a maximum of 82.1% (130 nm DFF). The memory-PUF benefiting the most from this technique is the 130 nm DFF-PUF; not only its area overhead reduction ranges from a minimum of 78% up to 82.1% (depending on the FE construction) but also its noise reduces from 28% down to 9%, its μ-BCHD increases from 0.43 up to 0.46 and its H_∞ increases from 0.61 up to 0.63, see Tables II and III. Similar improvements are obtained for both 65 nm DFF-PUF and 65 nm BK-PUF. Applying the noise reduction method for SRAM-based PUF systems reduces its area overhead ranging from a minimum of 31.6% up to 35.2% for 40 nm, while this range is 34.9% up to 43.1% for 130 nm. For 65 nm SRAM there is an increase in area ranging from of 5.2% up to 6.9%; however, both noise and min entropy are improved. The results show that the proposed noise reduction solution is attractive for all memory-based PUFs, particularly DFF and BK PUFs.

Regarding the cost of adding extra specific temperature/voltage ramp-up pairs, we estimate the following. Each new specific temperature/voltage ramp-up pair impacts only the controller design. Per new pair, a similar set of components as those used for 25 °C are required; i.e., one pMOS, one NAND, and two operational amplifiers. The areas of the pMOS and the NAND are very small when compared with that of the operational amplifiers. Therefore, we can estimate that the cost of adding a new specific temperature/voltage ramp-up pair is roughly the area overhead of two operational amplifiers; i.e., 125 GE = $\lceil (2 \times 0.0034\ \text{mm}^2)/(54.6\ \mu m^2) \rceil$, see Section VI-D. To have a better feeling of the number of extra temperature/voltage ramp-up pairs that make the noise optimized solution achieve the same area overhead of the nonoptimized, we carry out the following steps. First, from Fig. 15, we identify the PUF-system construction that has the least absolute area overhead reduction, i.e., 130 nm SRAM Golay-based, and calculate this value. Second, we divide the value from the first step by the GE of the extra components, i.e., 125 GE. We estimate that up to 12 new pairs can be added to the least reduced PUF system (in absolute terms), i.e., a total of 15 (12 plus the three pairs implemented in the previous section) fixed temperature/voltage ramp-up pairs. Therefore, we conclude that the proposed noise reduction solution is advantageous for a wide range of fixed temperature/voltage ramp-up pairs.

Finally, we would like to mention that any PUF size variation is mirrored by the helper data; helper data and PUF have the same size, see Fig. 7, hence, any increase or decrease in the PUF size due to noise reduction is intrinsically followed by the helper data. However, in this paper, we consider the helper data as being stored off-chip, and therefore, our results do not reflect its area reduction with PUF noise optimization.

B. Impact on Delay

In this type of industry we can easily tradeoff delay over higher reproducibility and higher uniqueness. Nonetheless, a delay analysis reveals the following. The total computational time, from power-up up to key reconstruction can be expressed by $\text{Total}_{\text{Delay}} = \text{Delay}_{\text{Sensors}} + \text{Delay}_{t_{\text{ramp}}} + \text{Delay}_{\text{Decoding}}$. The delay introduced by the sensors is negligible. The delay introduced by the t_{ramp} when compared to the original construction can be significant (depending on the temperature at which the

reconstruction is performed). However, with less noise, less PUF data is required. Therefore, the delay of the decoding is reduced. The number of iterations is constant for any given temperature and/ or ramp-up combination. The outcome of the tradeoff between the increase in t_{ramp} and decrease in decoding time is highly dependent on the frequency applied (as the t_{ramp} is fixed). However, as the key reconstruction phase is typically performed during power-up only, the overall impact of the method on the overall delay of the circuit is negligible. In other words, the area savings compensate for an eventual and discrete delay increase.

C. Generic Procedure

To investigate the noise reduction we performed measurements on ten voltage ramp-up times widely distributed (10 μs, 25 μs, 50 μs, 100 μs, 250 μs, 500 μs, 1 ms, 10 ms, 50 ms, and 500 ms). For any new technology node, type or architecture, new measurements would need to be performed (as an analytical model is too complex and unfeasible; among other issues, one would need to accurately describe the asymmetry between each memory cell). Obviously, a wider range of values with even more granularity would present more accurate results, however, it is more time consuming. Once the measurements are taken, they are analyzed by one of the proposed algorithms, hence, determining which temperature/voltage ramp-up time is optimal.

VIII. Conclusion

In this paper, we proposed a method for enhancing the reproducibility of memory-based PUFs based on adapting the voltage ramp-up time to the ambient temperature. The combined effect on PUF reproducibility has been evaluated using both circuit simulation and actual silicon measurements. The results are highly effective, showing a major decrease in worst-case PUF noise (up to 3× lower for particular PUFs) at extreme temperatures. The reproducibility enhancement is achieved while either maintaining or increasing the uniqueness. Furthermore, we investigated the relation between PUF noise and area overhead both for several types of memory-based PUFs and several memory-based PUF systems constructions. Our results show that when the PUF noise is reduced, the PUF size decreases up to 3× and that the footprint of the error correction system is also slightly reduced. Finally, we implemented a small and scalable circuit that adapts the voltage ramp-up time to the sensed ambient temperature. Overall, the implementation of the proposed method will result in a PUF-based key generator significantly smaller. The proposed solution is particularly attractive for less robust memory-PUFs, such as DFF and BK, boosting their competitiveness.

References

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

[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. Berlin, Germany: Springer, 2010, pp. 3–37.

[3] S. Katzenbeisser *et al.*, "PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon," in *Proc. CHES*, Leuven, Belgium, 2012, pp. 283–301.

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

[5] T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," in *Proc. ReConFig*, Quintana Roo, Mexico, 2010, pp. 298–303.

[6] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Proc. CHES*, Vienna, Austria, 2007, pp. 63–80.

[7] B. Skoric, P. Tuyls, and W. Ophey, "Robust key extraction from physical unclonable functions," in *Proc. ACNS*, New York, NY, USA, 2005, pp. 99–135.

[8] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

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

[10] R. Maes, A. Van Herrewege, and I. Verbauwhede, "PUFKY: A fully functional PUF-based cryptographic key generator," in *Proc. CHES*, Leuven, Belgium, 2012, pp. 302–319.

[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," in *Proc. CHES*, Leuven, Belgium, 2012, pp. 268–282.

[12] M. Hofer and C. Boehm, "An alternative to error correction for SRAM-like PUFs," in *Proc. CHES*, Santa Barbara, CA, USA, 2010, pp. 335–350.

[13] V. Vivekraja and L. Nazhandali, "Circuit-level techniques for reliable physically unclonable functions," in *Proc. HOST*, San Francisco, CA, USA, 2009, pp. 30–35.

[14] D. Forte and A. Srivastava, "On improving the uniqueness of silicon-based physically unclonable functions via optical proximity correction," in *Proc. DAC*, San Francisco, CA, USA, 2012, pp. 96–105.

[15] M. Bhargava, C. Cakir, and K. Mai, "Attack resistant sense amplifier based PUFs (SA-PUF) with deterministic and controllable reliability of PUF responses," in *Proc. HOST*, Anaheim, CA, USA, 2010, pp. 106–111.

[16] R. Kumar, H. K. Chandrikakutty, and S. Kundu, "On improving reliability of delay based physically unclonable functions under temperature variations," in *Proc. HOST*, San Diego, CA, USA, 2011, pp. 142–147.

[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 Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, Sep. 2009.

[18] M. Cortez, A. Dargar, S. Hamdioui, and G.-J. Schrijen, "Modeling SRAM start-up behavior for physical unclonable functions," in *Proc. IEEE Int. Symp. Defect Fault Toler. VLSI Nanotechnol. Syst.*, Austin, TX, USA, 2012, pp. 1–6.

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

[20] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA," in *Proc. HOST*, Anaheim, CA, USA, 2008, pp. 67–70.

[21] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic PUFs from flip-flops on reconfigurable devices," in *Proc. WISSec*, Eindhoven, The Netherlands, 2008, pp. 1–17.

[22] P. Simons, V. van der Leest, and E. van der Sluis, "Buskeeper PUFs, a promising alternative to D flip-flop PUFs," in *Proc. HOST*, San Francisco, CA, USA, 2012, pp. 7–12.

[23] Y. Su, J. Holleman, and B. Otis, "A 1.6pJ/bit 96% stable chip-ID generating circuit using process variations," in *ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, 2007, pp. 406–611.

[24] W. Zhao *et al.*, "Rigorous extraction of process variations for 65nm CMOS design," in *Proc. ESSDERC*, Munich, Germany, 2007, pp. 89–92.

[25] Predictive Technology Model. (2012). [Online]. Available: http://ptm.asu.edu/

[26] M. Cortez, S. Hamdioui, V. van der Leest, R. Maes, and G.-J. Schrijen, "Adapting voltage ramp-up time for temperature noise reduction on memory-based PUFs," in *Proc. HOST*, Austin, TX, USA, 2013, pp. 35–40.

[27] C. Bösch, J. Guajardo, A. R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," in *Proc. CHES*, vol. 5154. Washington, DC, USA, 2008, pp. 181–197.

[28] G. D. Forney, Jr., *Concatenated Codes* (Research Monograph), vol. 37. Cambridge, MA, USA: MIT Press, 1966.

[29] Taiwan Semiconductor Manufacturing Company Limited. (Oct. 2014). *65nm Technology Overview*. [Online]. Available: http://www.tsmc.com/english/dedicatedFoundry/technology/65nm.htm
[30] Taiwan Semiconductor Manufacturing Company Limited. (Oct. 2014). *40nm Technology Overview*. [Online]. Available: http://www.tsmc.com/tsmcdotcom/PRListingNewsAction.do?action=detail&language=E&newsid=2561
[31] Europractice. (Oct. 2014). *0.13um Technology Overview*. [Online]. Available: http://www.europractice-ic.com/technologies_TSMC.php?tech_id=013um
[32] M. Cortez, G. Roelofs, S. Hamdioui, and G. Di Natale, "Testing PUF-based secure key storage circuits," in *Proc. DATE*, Dresden, Germany, 2014, pp. 1–6.
[33] AMS. *0.35μ CMOS Technology Selection Guide*. [Online]. Available: http://www.ams.com/eng/Products/Full-Service-Foundry/Process-Technology/CMOS/0.35-m-CMOS-Technology-Selection-Guide
[34] C. Böhm and M. Hofer, *Physical Unclonable Functions in Theory and Practice*. New York, NY, USA: Springer, 2013
[35] Y. Tsividis and C. McAndrew, *Operation and Modeling of the MOS Transistor*, 3rd ed. New York, NY, USA: Oxford, 2011.
[36] J. Chang, A. A. Abidi, and C. R. Viswanathan, "Flicker noise in CMOS transistors from subthreshold to strong inversion at various temperatures," *IEEE Trans. Electron Devices*, vol. 41, no. 11, pp. 1965–1971, Nov. 1994.

Mafalda Cortez (S'12) received the M.Sc. degree in electrical and computers engineering—telecommunications, electronics and computers from the Faculdade de Engenharia da Universidade do Porto, Porto, Portugal. She is currently pursuing the Ph.D. degree with the Computer Engineering Laboratory, Delft University of Technology, Delft, The Netherlands, in collaboration with Intrinsic-ID B.V., Eindhoven, The Netherlands. During the M.Sc. degree, she did her graduation thesis at NXP Semiconductors Research, Eindhoven, the Netherlands, entitled "Electrical Characterization and Interpretation of Micro-Electro-Mechanical Systems Microphones With Spring Suspended Backplates.

She was an Invited Researcher with Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier Laboratory, Montpellier, France, to research novel secure design-for-testability schemes. Her current research interests include circuit design and modeling, hardware security, and secure IC test.

Said Hamdioui (M'99–SM'11) received the M.S.E.E. and Ph.D. degrees (both with Hons.) from Delft University of Technology (TUDelft), Delft, The Netherlands.

He was with Intel, Santa Clara, CA, USA, Philips Semiconductors Research and Development, Crolles, France, and Philips/NXP Semiconductors, Nijmegen, The Netherlands. He is currently co-leading dependable-nano computing research activities with the Computer Engineering Laboratory, TUDelft. He has consulted for several semiconductor companies. His current research interests include testability and design-for-test, reliability, hardware security, and emerging computation paradigms based on memristor technology. He published one book and co-authored over 130 conference and journal papers.

Dr. Hamdioui serves on the Editorial Board of the IEEE DESIGN AND TEST and the *Journal of Electronic Testing: Theory and Applications*. He is an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE-SCALE INTEGRATION (VLSI) SYSTEMS. He is strongly involved in the international test technology community and has delivered dozens of keynote speeches, distinguished lectures, and invited presentations and tutorials at major international forums/conferences and leading semiconductor companies. He is a member of Association for European Nanoelectronics Activities/ENIAC Scientific Committee Council.

Ali Kaichouhi is currently pursuing the M.Sc. degree in electrical engineering, track of microelectronics with the Delft University of Technology (TUDelft), Delft, The Netherlands.

He was with several companies as a Hardware Design/Network Engineer. He is a Support Engineer for IC-Design and Measurement with the TUDelft, where he researches on electronic design, support Cadence IC design kit technologies, Cadence layout design, electrostatic discharge, high voltage IC design, mixed signal IC design, RF IC design, layout verification in Cadence Assura, Mentor Graphics Calibre, Cadence Skill Programming, and Verilog-AMS.

Vincent van der Leest received the master's degree in electrical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands.

He is a Senior Project Leader with Intrinsic-ID B.V., Eindhoven, responsible for research and subsidy projects. He is regularly invited to teach lectures on physically unclonable functions (PUFs). His current research interests include PUFs, coding theory, and hardware security implementations. He has co-authored around 20 scientific publications in different security conferences and journals.

Roel Maes received the M.E.E. and Ph.D. degrees from Katholieke Universiteit Leuven, Leuven, Belgium, in 2007 and 2012, respectively.

He is a Hardware Security Engineer with Intrinsic-ID B.V., Eindhoven, The Netherlands, developing and integrating security architectures and solutions for innovative applications. He has co-authored over 25 papers in high-ranking security venues and has published a book on the topic of physically unclonable functions (PUFs). His current research interests include PUFs, information and coding theory, and security architectures in general.

Dr. Maes regularly performs reviews for the IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN and the *Journal of Cryptographic Engineering*. He is a Recurring Program Committee Member of Cryptographic Hardware and Embedded Systems and Design, Automation & Test in Europe Conference & Exhibition.

Geert-Jan Schrijen received the master's degree in electrical engineering from the University of Twente, Enschede, The Netherlands, in 2000.

In 2001, he joined the Security Group of Philips Research, Eindhoven, The Netherlands, where he researched on digital rights management, low-power authentication protocols, private biometrics, and physical unclonable functions. He was a Senior Algorithm Designer with Intrinsic-ID B.V., Eindhoven, where he focused on the development of signal processing algorithms and security architectures for hardware-intrinsic key storage systems. In 2011, he was appointed as a VP Engineer with Intrinsic-ID B.V., where he is currently the Head of the Engineering Team that is responsible for hardware and software development.