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 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; hence, an overall more efficient implementation. This paper proposes novel insight and resulting method 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, 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 PUF noise up to $3 \times$ 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 $3 \times$ higher reproducibility.

Index Terms—Noise Reduction, Memory-based PUF, Adapter Circuit, 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, 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 (FE) [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 FE byproduct, 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 (ECCap) 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, such as 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 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 PUF responses noise level have been proposed, either 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, an extension of our work 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 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 [17–19]. 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, 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 fuzzy extractor 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 the 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 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 [6,20–23] 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 \((Q_5, Q_6)\), the bitline (BL), the complement bitline (BLB) and the wordline (WL). 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 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.

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 FE [8,9], which typically consists of two phases:

(a) 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 a FE byproduct. Finally, the Helper Data is stored in an external NVM; hence, it becomes public information.

(b) Reconstruction: the earlier enrolled Cryptographic Key is reliably recovered. First, a PUF measurement is taken and used as PUF Response (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 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. PUF response 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 PUF responses, the more efficient the overall PUF-based key generation system can be implemented.

2) Uniqueness: To generate a secure key, a FE requires PUF response unpredictability, even if 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

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 PUF response is generated by an individual SRAM cell. Fig. 3 shows the SRAM fingerprint generation schematic used in our simulations. [17,18] 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 $\mu = \text{standard NMOS } V_{th}$ 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 45nm 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:

- **Voltage ramp-up time:** $4 \times t_{ramp}$ (10µs, 50µs, 90µs and 130µs),
- **Temperature:** $3 \times Temp$ ($-40^\circ C$, $+25^\circ C$, $+85^\circ C$) and,
- **Measurements:** $20 \times Meas$ (each with a random seed).

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

D. Simulation Results and analysis

Fig. 4 shows the results of maximum FHD ($\text{max FHD}$) 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 $+25^\circ C$ with $t_{ramp}$ of (a) 10µs, (b) 50µs, (c) 90µs and (d) 130µs; the enrollment conditions are given between ‘[]’ in the figure. From Fig. 4(a) it can be seen that for $Temp$ below the enrollment, $\text{max FHD}$ 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 $+85^\circ C$, key-reconstruction with 10µs generates the lowest $\text{max FHD}$ while at $-40^\circ C$, that is true for 90µs.

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: (a) thermal noise $ThN$, (b) Flicker noise $FN$ and (c) shot noise $SN$ [36,37]. During the enrollment phase, we are in fact establishing a noise level reference, i.e.,

$$TN = ThN + FN + SN$$ (1)

where $TN$ is the total noise. First, $ThN$ is related to the scattering of carrier charges in thermal motion, and is di-
directly 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 [36,37]. Second, FN, also known as $1/f$ noise, is related to trapping and releasing charges near the Si-SiO2 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 [36,37]. 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 [36,37]. It is worth noting that ThN and FN have much larger impact than SN in the frequency range considered [38]. 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 ($-40^\circ C$, $+25^\circ C$ and $+85^\circ C$) and for ten different $t_{ramp}$ varying from $10\mu s$ up to $500\mu s$. In case of 40nm SRAM, the shortest possible $t_{ramp}$ is $50\mu s$ 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.
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 second.
6) Steps 3 to 5 are repeated 9 times (i.e. 10 measurements per PUF per temperature per $t_{ramp}$).
7) Change $t_{ramp}$ and repeat steps 3 to 6 (until all values of $t_{ramp}$ have been tested for this temperature).
8) Change temperature and repeat steps 3 to 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 PUF response (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 (a) the average between-class Hamming distance ($\mu$-BCHD) [17], and (b) the estimated min-entropy ($H_\infty$) [17] of the measured responses. Note that for key storage application (as described in Section II) only the uniqueness of the enrollment PUF response is critical, as it is from this response that the cryptographic key is derived. $\mu$-BCHD is calculated as follows:

TABLE I: Description of devices used in validation.

<table>
<thead>
<tr>
<th>Technology</th>
<th># ICs</th>
<th># PUF inst. / IC</th>
<th>Total # PUF inst.</th>
</tr>
</thead>
<tbody>
<tr>
<td>40nm LP</td>
<td>40</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>65nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40nm LP</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

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.
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 second.
6) Steps 3 to 5 are repeated 9 times (i.e. 10 measurements per PUF per temperature per $t_{ramp}$).
7) Change $t_{ramp}$ and repeat steps 3 to 6 (until all values of $t_{ramp}$ have been tested for this temperature).
8) Change temperature and repeat steps 3 to 7.
Optimally, the obtained distribution should be approximately Gaussian and \( \mu \)-BCHD should be very close to 50\% [17]. \( H_\infty \) is used to evaluate the intrinsic unpredictability of PUF responses. \( H_\infty \) is a pessimistic measure of a random variable unpredictability [8]. We estimate \( H_\infty \) of the responses of a particular PUF type by considering the following model: each PUF response 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 \( H_\infty \) is simply the summation of \( H_\infty \) of each bit. We express \( H_\infty \) as the average \( H_\infty \) per bit in a response value, by dividing the total \( H_\infty \) of the response by its length. Optimally, \( H_\infty \) of a PUF response bit should be close to 1. Note that, due to the limited number of measured PUF instances, the obtained estimations of \( H_\infty \) could be lower than the actual PUF responses \( H_\infty \).

C. Optimization Algorithms

The silicon measurements have the objective to investigate the use of \( t_{ramp} \) as a technique for increasing memory-based PUF response 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 \( H_\infty \). 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 analysed 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\°C, 25\°C and 85\°C) are represented on the y-axis and x-axis, respectively, whereas the max FHD is represented by color. These values are obtained using \( t_{ramp} \) from 10\( \mu s \), which is the shortest feasible \( t_{ramp} \) for each PUF, except for 40nm LP SRAM PUF where the shortest feasible \( t_{ramp} \) is 50\( \mu s \), up to 500\( \mu s \). The max FHD (noise) is determined using 10 response measurements per PUF per temperature per \( t_{ramp} \).

Fig. 5(a) reveals a clear convergence pattern towards 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 3 (i.e., \( t_{ramp} \) at 50\( \mu s \)), for -40\°C max FHD decreases until \( t_{ramp} \) increases thereafter, while for both 25\°C 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 65nm DFF PUF), 8\% at +25\°C (for the 65nm DFF PUF), and 28\% at +85\°C (for the 130nm DFF PUF). Regarding uniqueness, although a truly fair comparison is not possible due to limited available devices per technology node and PUF type, the 65nm DFF PUF has the lowest \( \mu \)-BCHD = 0.37 and \( H_\infty = 0.40 \).

In addition, to investigate whether the observed convergence towards 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 10 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 longer \( t_{ramp} \), whereas for high temperatures this is realized with shorter \( t_{ramp} \). For example, the maximum noise for 65nm LP DFF PUF at -40\°C with \( t_{ramp} = 10\mu s \) for both enrollment and reconstruction is originally 28\%. However, if the optimized \( t_{ramp} \) is used both at enrollment (500\( \mu s \) at +25\°C) and at reconstruction (50\( \mu s \) 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, \( \mu \)-BCHD and \( H_\infty \) are deteriorated for some PUFs (e.g., 130nm SRAM PUF), while they are significantly improved for others (e.g., 65nm DFF PUF).

2) Uniqueness optimization: Table IV reports the uniqueness optimization algorithm results; it shows (a) the \( t_{ramp} \) at enrollment that maximizes uniqueness and (b) the \( t_{ramp} \) for the

### TABLE II: Measurement results without optimization.

<table>
<thead>
<tr>
<th>Technology</th>
<th>PUF</th>
<th>( t_{ramp} )</th>
<th>Maximum noise FHD ( \mu )-BCHD</th>
<th>( H_\infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>40nm LP</td>
<td>SRAM</td>
<td>50( \mu s )</td>
<td>23% 6% 20% 0.50 0.73</td>
<td></td>
</tr>
<tr>
<td>65nm LP</td>
<td>DFF</td>
<td>10( \mu s )</td>
<td>8% 6% 8% 25% 0.37 0.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BK</td>
<td>10( \mu s )</td>
<td>10.5% 4.5% 20% 0.48 0.75</td>
<td></td>
</tr>
<tr>
<td>130nm LP</td>
<td>SRAM</td>
<td>10( \mu s )</td>
<td>13% 6% 12% 0.47 0.66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DFF</td>
<td>10( \mu s )</td>
<td>16.5% 5% 28% 0.43 0.61</td>
<td></td>
</tr>
</tbody>
</table>
other temperatures that results in the lowest maximum noise (w.r.t. the $t_{ramp}$ Selected for enrollment). Uniqueness indicators $\mu$-BCHD and $H_\infty$ are at least as high as the originals for 40nm and 130nm SRAMs, and for the remaining devices these indicators are higher than the original indicators. The uniqueness optimization algorithm clearly leads to significant improvements in $\mu$-BCHD and $H_\infty$ for DFF and buskeeper 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 65nm SRAM PUF, the maximum noise at $-40^\circ$C is even worse than the measurements without optimization. Reason for this is that $t_{ramp}$ at enrollment ($+25^\circ$C) is very long and the algorithm is unable to find a corresponding longer $t_{ramp}$ at $-40^\circ$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 PUF response 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 '1's 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 on Table II while either maintaining or increasing the uniqueness indicators. Inspecting the silicon results w.r.t. reproducibility and uniqueness reveals the following:

- 40nm and 65nm 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.
- 130nm SRAM PUFs benefit from applying the uniqueness optimization algorithm, as the noise is reduced while the uniqueness is maintained.
- Buskeeper 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 PUF response uniqueness, the proposed algorithm also decreases the noise at −40°C 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 fuzzy extractor 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

A Fuzzy Extractor (FE) is a fundamental component of a PUF-based key storage system (see Fig. 2); it has two main functions. (a) Information reconciliation: it uses the helper data to correct errors on the measured PUF response; and (b)
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 PUF response.

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 the noise and the 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. 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 [29]. The aforementioned FE constructions owe their popularity to their area overhead efficiency when compared with their BCH counterparts, while delivering the same error correction efficiency [29,30]. For this reason our study 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: (a) efficiently reconstructs a valid cryptographic key from a true PUF device (the one used for enrollment) under various conditions and, (b) 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 PUF response 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 PUF response of device B is such that it is mistakenly corrected to the PUF reference response 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 PUF response and ECCap. Let’s consider the system shown in Fig. 7. During Key Reconstruction, a 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 [l/k] decoding iterations. The probability that a true key is not reconstructed can be expressed as [28]:

$$\text{FRR} = 1 - (1 - \text{PE}_{Xcode})^{\text{iterations}}$$

(2)

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

$$\text{PE}_{Xcode} = \sum_{i=0}^{s} \binom{s}{i} \text{PE}_{rep}^i (1 - \text{PE}_{rep})^{s-i} = 1 - \sum_{i=0}^{s} \binom{s}{i} \text{PE}_{rep}^i (1 - \text{PE}_{rep})^{s-i}$$

(3)

Fig. 8: False Rejection Rate and False Acceptance Rate [27]
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2015.2422844, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

In addition, we perform the simulation for the following values:

\*\*\*\*

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) 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 [6,29], hence, \([128/0.75]=171\) bits are required [29].

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

1) 51 different \( \epsilon \); we sweep \( \epsilon \) from 5% up to 30% with a step of 0.5%.
2) 3 different combinations of \( s \) and \( k \); these reflect 3 FE constructions: (a) Golay-based with \( \{ s, k \} = \{24,12\} \), (b) RM16-based with \( \{ s, k \} = \{16,5\} \) and (c) RM8-based \( \{ s, k \} = \{8,4\} \).

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., 130nm 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 40nm and 130nm a quite saving is achieved for all FE constructions, almost no saving is realized for 65nm 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.

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 PUF response error probability \( \epsilon \). From the figure we can make the following conclusions.

1) Reduction in noise \( \epsilon \) significantly reduces the required PUF size and \( n \). Regardless of the FE construction, the lower \( \epsilon \), 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 \( 3 \times \) causes a 2.6 \( \times \) 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 \( \times \) when \( \epsilon \) reduces from 15% to 5%. However, this is only 2.3 \( \times \) for RM8-based. Moreover, overall, Golay-based PUF system is the one with smaller PUF size and \( n \) for any given \( \epsilon \).

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 (\( \approx \epsilon \)), 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).

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.
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2015.2422844, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

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 65nm SRAM PUF) up to 20 kGE (for the 65nm DFF PUF); GE (Gate Equivalent) is a technology node independent metric of area that denotes the area of a 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 [31–33].

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., $t_{ramp}$, 10μs at +85°C, 1ms at +25°C and 500ms at −40°C.

In short, the requirements are:
- Low area overhead (up to budget).
- Output $t_{ramp} = 10μs$ at +85°C, $t_{ramp} = 1ms$ at +25°C and $t_{ramp} = 500ms$ 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 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, a voltage ramp-up regulator and controller. The general design of these extensions are 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.

C. Implementation

Fig. 13 shows the schematic of the circuit; where part (a) of the figure depicts the embedded temperature sensor, part (b) the controller and part (c) the voltage ramp-up regulator. The circuit is implemented in 0.35μm, 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 $V_{control}$. 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
It is worth emphasizing that the proposed circuit generates from the continuous range of voltage ramp-up times, we fix the temperature we perform an AND (AND0) operation as MP3 below (for $\mu^\circ C$ only one comparison is required as we only need to make sure that the $V_{temp}$ is above (for +85$^\circ C$) and below (for $-40^\circ C$) the reference voltage of the respective temperature. For intermediary temperatures (i.e., +25$^\circ C$) 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 a 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., $10\mu s$ at $+85^\circ C$, $1ms$ at $+25^\circ C$ and $500ms$ at $-40^\circ C$). 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 500ms down to $10\mu s$, as the temperature increases from $-40^\circ C$ up to $+85^\circ C$; 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 $-40^\circ C$ a $t_{ramp}$ of 500ms, at $+25^\circ C$ a $t_{ramp}$ of 1ms and at $+85^\circ C$ a $t_{ramp}$ of $10\mu s$, as required. Moreover, as predicted, the voltage ramp-up time decreases continuous and monotonically from 500ms down to $10\mu s$, as the temperature increases from $-40^\circ C$ up to $+85^\circ C$; e.g., at $-30^\circ C$ the circuit outputs a $t_{ramp}$ of 358ms, while at $75^\circ C$ it outputs a $t_{ramp}$ of $12.6\mu s$. 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 $563.36\mu m^2$ ($22.4\mu m \times 25.15\mu m$) which is fixed regardless of the resolution of the system and it is easily implementable in other technology nodes. The controller, as designed, outputs one of the three reference voltages ($V_{m40} = 2.65\, V$, $V_{25} = 2.53\, V$ or $V_{85} = 2.38\, V$). It has an area of $0.014mm^2$ ($71.5\mu m \times 204.5\mu m$), which 90% corresponds to the area of the operational amplifiers (area of one operational amplifier $0.0034mm^2$). The controller is easily implementable in other technology nodes. The temperature sensor outputs a voltage with a linear relation with the temperature; $V_{temp} = 1.32V$ at $-40^\circ C$, $1.47V$ at $+25^\circ C$ and $1.61V$ at $+85^\circ C$, which results in a resolution of $2.5mV/\circ C$. The temperature sensor has an area of $169.035\mu m^2$ ($8.85\mu m \times 19.1\mu m$). 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 $0.015mm^2$ ($70.9\mu m \times 214.75\mu m$).

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 [35]; 0.015nm$^2$ corresponds to 275 GE ($= \frac{0.015nm^2}{54.6\mu m^2}$), where 54.6 $\mu m^2$ corresponds to the area of a NAND2 cell in 0.35nm [35]). 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, 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 work, 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 ($c$) 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 [34], the Repetition code hardware implementation comprises a counter, which counts up to $n$ (length of the repetition code). The higher $n$ 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% (40nm SRAM) up to a maximum of 82.1% (130nm DFF). The memory-PUF benefiting the most from this technique is the 130nm 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 $\mu$BCHD increases from 0.43 up to 0.46 and its $H_\infty$ increases from 0.61 up to 0.63, see Tables II and III. Similar improvements are obtained for both 65nm DFF-PUF and 65nm 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 40nm, while this range is 34.9% up to 43.1% for 130nm. For 65nm SRAM there is an increase in area ranging from of 5.2% up to 6.9%; however, both the noise and the 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., 1 PMOS, 1 NAND and 2 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 = $\left\lceil \frac{2 \times 0.0034nm^2}{54.6\mu m^2} \right\rceil$, 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 non-optimized, 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., 130nm 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 our study, 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 trade-off delay over higher reproducibility and higher uniqueness. Nonetheless, a delay analysis reveals the following. The total computational time, from power-up to key reconstruction can be expressed by \( Total_{delay} = Delay_{sensors} + Delay_{ramp} + Delay_{decoding} \). 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 combination. The outcome of the trade-off 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\( \mu s \), 25\( \mu s \), 50\( \mu s \), 100\( \mu s \), 250\( \mu s \), 500\( \mu s \), 1ms, 10ms, 50ms, 500ms). 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 \( 3 \times \) 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 \( 3 \times \) 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.

ACKNOWLEDGMENTS

The work performed here was partially supported by the European Commission through the FP7 programme under contract 284833 PUFFIN and partially by the Dutch “Point One Program” under the RATE project (PNU09C09).

REFERENCES

Said Hamdioui received the MSEE and PhD degrees (both with honors) from Delft University of Technology (TUDelft), Delft, The Netherlands. He is currently co-leading dependable-nano computing research activities within the Computer Engineering Laboratory of TUDelft. Prior to joining TUDelft, Hamdioui worked for Intel CA, for Philips Semiconductors R&D France, and for Philips/ NXP Semiconductors the Netherlands. His research interests include Testability and Design-for-Test, Reliability, Hardware Security and emerging computation paradigms based on memristor technology. Hamdioui published one book and co-authored over 130 conference and journal papers. He has consulted for many semiconductor companies. 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. Hamdioui is a Senior member of the IEEE; he serves on the editorial board of IEEE Design & Test, and of the Journal of Electronic Testing: Theory and Applications (JETTA). In addition, he is Associate Editor of IEEE Transactions on VLSI Systems (TVLSI), and a member of AESENS/ENIAC Scientific Committee Council (AESENS =Association for European Nanoelectronics Activities).

Ali Kaichouhi is pursuing his M.Sc. in Electrical Engineering, track of Microelectronics at Delft University of Technology (TU Delft). At the same time, he is working as Support Engineer for IC Design and Measurement at the TU Delft. The work relates to electronic design, support Cadence IC DesignKit Technologies, Cadence Layout Design, ESD, High Voltage IC design, Mixed Signal IC design, RF IC design, Layout verification in Cadence Assura, Mentor Graphics Calibre, Cadence Skill Programming, Verilog-AMS, etc. Prior to joining for several companies as hardware design engineer.

Vincent van der Leest is Senior Project Leader at Intrinsic-ID, responsible for research and subsidy projects. Over the last years he has co-authored around 20 scientific publications at different security conferences and in journals. His research is oriented toward PUFs, coding theory and hardware security implementations. He holds a masters degree in Electrical Engineering from Eindhoven University of Technology and is regularly invited to teach lectures on Physically Unclonable Functions.

Roel Maes is a Hardware Security Engineer at Intrinsic-ID, 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 PUFs. He regularly performs reviews for IEEE TCAD and Journal of Cryptographic Engineering, and he is a recurring program committee member at CHES and DATE. His research is focused on PUFs, information and coding theory, and security architectures in general. He holds a MEE degree from KU Leuven.

Geert-Jan Schrijen received his Masters degree in Electrical Engineering from the University of Twente in December 2000 on the topic of ‘active noise cancellation with multiple loudspeakers. In April 2001 he joined the security group of Philips Research in Eindhoven where he worked on digital rights management, low-power authentication protocols, private biometrics and Physical Unclonable Functions (PUFs). Work on the latter topic resulted in the publication of his PhD thesis in 2008. As a senior algorithm designer within Intrinsic-ID, Geert-Jan focused on the development of signal processing algorithms and security architectures for hardware-intrinsic key storage systems. In January 2011 he was appointed VP Engineering and is now heading the engineering team that is responsible for hardware and software development within Intrinsic-ID.

Mafalda Cortez is currently pursuing her Ph.D. at the Computer Engineering Lab from the Delft University of Technology in collaboration with Intrinsic-ID. In addition, she was an invited researcher at the LIRMM laboratory, in Montpellier, France, to research novel secure DFT schemes. Her research interests include Circuit Design and Modeling, Hardware Security and Secure IC Test. She received her M.Sc. degree in Electrical and Computers Engineering - Telecommunications, Electronics and Computers from the Faculdade de Engenharia da Universidade do Porto (FEUP), Portugal. During her master’s education, she did her graduation thesis at NXP Semiconductors Research, in Eindhoven, the Netherlands, entitled ‘Electrical Characterization and Interpretation of MEMS Microphones with Spring Suspended Backplates’.