This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

Improved Generation of Identifiers, Secret Keys, and Random Numbers from SRAMs

Iluminada Baturone, Miguel A. Prada-Delgado, and Susana Eiroa

Abstract*—This paper presents a method to simultaneously improve the quality of the identifiers, secret keys, and random numbers that can be generated from the start-up values of standard Static Random Access Memories, SRAMs. The method is based on classifying memory cells after evaluating their startup values at multiple measurements in a registration phase. The registration can be done without unplugging the device from its application context, and with no need for a complex laboratory setup. The method has been validated experimentally with standard low-power SRAM modules in two different Application Specific Integrated Circuits, ASICs, fabricated with 90-nm TSMC technology. The results show that with a simple registration the length of the identifiers can be reduced by 45%, worst-case bit error probability (which defines the complexity of the error correcting code needed to recover a secret key) can be reduced by 64%, and the worst-case minimum entropy value is improved, thus reducing the number of bits that have to be processed to obtain full entropy by 81%. The method can be applied to standard digital designs by controlling the external power supply to the SRAM using software or by incorporating simple circuitry in the design. In the latter case, a module for implementing the method in an ASIC designed in 90-nm TSMC technology occupies an active area of 42,025 μm^2.

***Index Terms*—SRAMs, PUFs, random numbers, hardware security**

I. INTRODUCTION

STATIC memory cells have two stable states. They store the logic '0' or '1' that is written in them when they are powered up, and lose information when they are powered down. If they are powered up and no data is written, they reach logic '0' or '1' in a way that can be difficult to predict, to model mathematically, and to clone physically, making them behave as Physical Unclonable Functions, PUFs. It has been observed that many of the memory cells in a memory reach the same start-up values every time that memory is powered up and that those values are not the same as the start-up values reached by the cells of other memories [1]-[10]. This property has been exploited to construct identifiers (IDs) which cannot be generated by counterfeit memories and are more difficult to copy because they are generated on the fly rather than being stored. Static memory cells consist of cross-coupled circuits with latches, flip-flops, NOR gates, inverters, etc. Cells based on cross-coupled latches [1] and flip-flops [2] have been employed to identify FPGAs. In [3], cells based on NOR gates were specifically designed to identify ASICs. The memory cells of SRAMs, which consist of cross-coupled inverters, were first proposed in [4] to identify integrated circuit wafers and dies and were later studied by many authors [5]-[10]. This paper focuses on exploiting the standard SRAMs available in many digital circuits for security purposes.

Ideally, the two inverters of the SRAM cells should be as identical as possible, since this improves the power and speed characteristics of the memory. However, mismatching produced by fabrication process variability is unavoidable and the two inverters will therefore hardly ever behave in the same way. If the cell is started-up and no value is written, any minor difference between the inverters will cause one of them to start conducting before the other. Due to the amplifying effect of each inverter acting on the output of the other, this will bias, or skew the cell towards '0' or '1'. The work in [5] classifies cells into two basic types: "skewed cells", displaying high degree of bias and therefore more likely to evolve to one of the stable states, and "neutral cells" that do not have a strong tendency to any state.

The big problem when generating identifiers and secret keys is that neutral cells change their start-up value from one generation to another. This is known as the bit flipping problem [6]-[12]. In identification applications, some bit flipping is allowable provided that the identifiers generated by genuine and fake memories are sufficiently different. However, cryptographic keys cannot vary from one realization to another (if they could, it would be impossible to encode and/or authenticate messages correctly). Bit flipping must therefore be removed completely to construct secrets from SRAM responses. For this purpose, Helper Data Algorithms (HDAs) - in particular Code Offset-based HDAs as described in [13] - have been employed. The drawback to this solution is the use of heavy error correction codes (ECCs) [6], [7]. As

Manuscript received March 13, 2015; revised June 28, 2015; accepted August 5, 2015. Manuscript received in final form August 18, 2015. This work was supported in part by RTC-2014-2932-8 and TEC2014-57971-R projects from *Ministerio de Economía y Competitividad* of the Spanish Government (with support from the PO FEDER-FSE). The work of Miguel A. Prada-Delgado was supported by *V Plan Propio de Investigación* through the University of Seville, Seville, Spain.
Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.
Iluminada Baturone and Miguel A. Prada-Delgado are with the Microelectronics Institute of Seville (IMSE-CNM), CSIC and University of Seville, Seville, Spain (e-mail: lumi@imse-cnm.csic.es, prada@imse-cnm.csic.es).
Susana Eiroa is with ALTER Technology TÜV NORD, Seville, Spain (e-mail: Susana.eiroa@altertechnology.com).

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

reported in [7], [9], the percentage of cells showing bit flipping increases if there are variations in the power supply voltage and increases even more if there are variations in the operating temperature.

Many techniques have been reported aimed at minimizing bit flipping. Some of them resort to the full custom VLSI design of memory cells [14]-[15]. In [14], NMOS write drivers and PMOS switches are added to the memory cells while in [15] circuitry is included to add an additional voltage source at the gate of one of the NMOS transistors in the memory cell and evaluate whether the threshold voltage mismatch between the NMOS transistors exceeds a carefully pre-established threshold. In any case, such approaches are more costly than using standard SRAM cells. The problem of using standard SRAM cells is that the influence of temperature and power supply voltage on the final start-up values is difficult to model, although some interesting studies have been carried out in this regard (see [16]). Several authors have proposed experimentally characterizing the behavior of the start-up values at multiple environmental temperatures (at least three temperatures), by using a temperature chamber or a temperature forcing system to control the operating temperature of the memory [17], [18]. As an alternative to this exhaustive characterization of many operating conditions, [12] proposed verifying two corner conditions (high-temperature low-power-supply-voltage and low-temperature low-power-supply-voltage), together with a neighborhood analysis within a memory word to select the most suitable cells for generating identifiers or keys. However, those corner conditions are not common to all SRAMs and neighborhood influence also depends on the particular SRAM in question (for example, strong location-based correlations are reported in [19] but not in other works). Furthermore, startup values are influenced not only by the final supply voltage value but also by the voltage ramp-up time (i.e., the time it takes to reach the operational supply voltage after power-on) [9]. The work in [20] therefore proposed characterizing the SRAM in terms both of power supply ramp-up times and temperatures using the appropriate laboratory equipment in a registration phase and then matching the ramp-up time to the ambient temperature when the start-up values are needed (this can be implemented using a voltage ramp-up regulator, an embedded temperature sensor, an analog to digital converter and a controller). To reduce the percentage of bit flipping, the work in [11] analyzed three reliability enhancing techniques: directed accelerated aging, activation control (which controls the power supply waveform shape, in particular the ramp-up time), and multiple evaluations under nominal operating conditions to generate a "soft" version of the SRAM startup values (the multiple evaluations can then be combined using majority vote or saturating arithmetic to form the final "hard" response). The best results were obtained using the last technique, and it is therefore this technique that this paper explores in greater depth.

All the above mentioned works focus on improving the generation of identifiers or secret keys but they do not propose anything regarding random numbers. It must be remembered that while skewed cells are the most suitable for generating reliable identifiers and secret keys, they should not be used to generate sequences of random numbers because their start-up values are repeated over and over again. In contrast, neutral cells, which should not be used to generate identifiers and secret keys, are the best cells for generating random numbers.

The metastability of cross-coupled circuits has been widely exploited to produce true random number generators (TRNGs). The earliest solutions, which were based on latches and flip-flops, appeared in [21], [22]. SRAM start-up values were first used as a source of entropy in [5]. Since the entropy provided by standard SRAMs is not very high (because they have many skewed cells), the works in [5] and [23] used a hash function to condense many bits into a much shorter bit string, ensuring full entropy. As in the case of identifiers or secret keys, some techniques have also resorted to the full custom VLSI design of memory cells to generate random numbers, although for this purpose the design objective is just the opposite: to hold the cell at the metastable point or to evaluate the quality of the cell metastability [22], [24]. In [22], a negative feedback loop implemented with a switched capacitor network is employed. In [24] additional circuitry included a completion detector, a time-to-digital converter, and a control system employing statistical information from a set of measurement samples. The work in [25] proposed the inclusion of digital signal processing circuitry to implement 8 NIST (National Institute of Standards and Technology) tests suitable for evaluating the quality of the random numbers as they are generated. All these approaches focus on improving the generation of random numbers, but do not propose anything regarding identifiers and secret keys.

The method presented in this paper offers a good tradeoff between implementation cost and improvements in the generation of identifiers, secret keys and random numbers - security primitives required by many cryptographic applications. To the best of the authors' knowledge, no other method has been proposed to improve the generation of all these primitives simultaneously. The idea is to classify the memory cells of the standard SRAMs available in many digital designs into two disjoint sets, one suitable for generating identifiers or secret keys, and the other suitable for generating random numbers. No complex laboratory setup is required for such classification. The method can be implemented easily with the memory embedded in its application context, either by adding simple circuitry to the digital design or by executing simple software. It does not need to be implemented by specialized vendors, or in the factory where the memory is manufactured.

The paper is structured as follows. Section II describes the methodology used to characterize SRAMs, defining the performance metrics and the evaluation strategy followed. Section III contains experimental results for reliability and entropy obtained from standard TSMC 90-nm SRAMs included in two different ASICs. The results obtained with and without classifying the cells are compared and discussed to support the choice of the proposed method. Section IV describes how the method is applied and uses experimental

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

results to validate the advantages of its application to generate identifiers, secret keys, and random numbers. A VLSI module was designed in TSMC 90-nm technology to illustrate, together with the SRAMs analyzed, how an ASIC with embedded SRAMs can incorporate the method. The advantages of the method for countering aging are summarized at the end of Section IV. Finally, conclusions are given in Section V.

II. Methodology to Characterize SRAMs

A. Performance metrics

1) Reliability

The use of start-up values to identify SRAMs consists of two steps, registration and verification. In the registration step, response R_i resulting from the concatenation of the start-up values of n memory cells of the SRAM is stored as the template of that memory. In the verification stage, the start-up values of the n cells are again measured, obtaining response R_j. Ideally, the responses should be the same. However, some bit flipping will inevitably occur. If m responses generated by the same n cells at different times are considered, an estimate of the reliability of the identification is given by the maximum fractional Hamming distance, $max_{IntraHD}$, between all the possible pairs of responses. This is defined as follows:

$$\max_{IntraHD} = \max_{\substack{i=1,\ldots,m-1 \\ j=i+1,\ldots,m}} \left[\frac{HD(R_i, R_j)}{n}\right] \cdot 100 \quad (1)$$

In the ideal situation of 100% reliability, all the responses should be the same, and $max_{IntraHD}$ will therefore be zero.

For secret key generation, the Code Offset-based Helper Data Algorithms, as described in [13], consist of two steps. In the initialization step, a response, R, is provided by the SRAM and a codeword, c, is randomly chosen from an Error Correcting Code (ECC). The XOR of R and c forms the code offset that is stored as helper data, $W = R \oplus c$. In the key generation step, a new response, R', is provided by the SRAM. The XOR of R' and the data stored, $c' = R' \oplus W = R' \oplus R \oplus c$, enters the decoder of the error correcting code to recover c and then the initial response R, which is used as a seed for a cryptographic key generation algorithm.

The worst-case probability that a bit in the SRAM responses may change (the bit error or bit flipping probability p) can be estimated by the $max_{IntraHD}$ defined in Equation (1). With a probability of $1-p$, the bit in the responses does not change. Assuming that all bits in the responses are independent, the probability of exactly t errors occurring in n bits is given by a binomial distribution, as follows:

$$P(t) = \binom{n}{t} \cdot p^t \cdot (1-p)^{n-t} \quad (2)$$

Hence, the probability that a string of n bits contains more than t errors is given by:

$$P_{total} = 1 - \sum_{i=0}^{t} \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i} \quad (3)$$

The authors in [13] and [26] proposed the use of the model described above to select the most suitable ECC according to bit error probability, p. Given p (estimated by $max_{IntraHD}$) and given an ECC (with n-bit codewords and capacity to correct up to t errors), the capability of the ECC to achieve a given P_{total} can be evaluated with Equation (3). The number of errors to be corrected and, hence, the complexity of the ECC, increases as the value of $max_{IntraHD}$ gets higher.

2) Minimum entropy

Minimum entropy summarizes the adequacy of n memory cells to generate random numbers. According to NIST recommendations [27], minimum entropy measures the worst case of uncertainty in a random variable. If the random variable is the start-up value observed at the i-th memory cell and p_{imax} is the maximum probability of taking logic value '0' or '1', the minimum entropy of the cell as a binary source of randomness is:

$$(H_{\min})_{cell} = -\log_2(p_{i\max}) \quad (4)$$

Assuming that the n memory cells have independent start-up values, the minimum entropy of the n-bit sequence (given as a percentage) is:

$$H_{\min} = -\frac{1}{n}\sum_{i=1}^{n} \log_2(p_{i\max}) \cdot 100 \quad (5)$$

For example, if the reliability of the n cells is 100% then the minimum entropy is 0%. In contrast, if the reliability is 0% and the p_{imax} of the n cells is 0.5 then the minimum entropy is 100%. The method described above is used in [23] to evaluate the minimum entropy of SRAMs. The method described in [5] assumes that each byte, instead of each bit, of the SRAM is an independent source. Hence, both methods are similar if no correlation exists between the bits of a specific byte. In any case, correlation between bits should be measured to test whether the assumption for Equation (5) is valid.

3) Stable and unstable cells

One way of evaluating whether n memory cells are adequate to generate IDs and secret keys is to measure their start-up values after several power-ups under different operating conditions. The S cells that always provide the same start-up value are adequate to generate IDs while the other U cells should not be used for this purpose ($n = S+U$). Since the IDs and secret keys are to be reproduced over varying operating conditions, the A cells that are always labeled as S for all the conditions are very adequate to generate them while the B cells that are always labeled as U always introduce bit flipping. There are also C cells, which are stable under certain conditions and unstable under others, so that $n = A + B + C$. Reliability depends on the percentage of stable cells; that is to

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

say, the cells that never show bit flipping. In contrast, the percentage of unstable cells is related to the capability to generate random numbers:

$$Percentage\ OfStableCells = \frac{A}{n} \cdot 100 \quad (6)$$

$$Percentage\ OfUnstable\ Cells = \frac{B}{n} \cdot 100 \quad (7)$$

Figure 1 shows the flowchart of the classification process.

4) Independence of sequences

Given two sequences of *n* bits, *R* and *R'*, provided by *n* memory cells of a SRAM, the way to measure their degree of similarity is to evaluate the number of *t* bits that are different in both sequences. This is given by their Hamming distance.

In the ideal situation of 100% independence, the comparison between each pair of bits in the sequences should be essentially a Bernoulli trial, which takes value '1' (the bits are different) with probability *p* (and a value of '0' with probability *q*=1-*p*). In addition, any given bit in the sequences should be equally likely to be '1' or '0', i.e., the sequences should be uniform to be unpredictable as commented below. Hence, ideally *p*=*q*=0.5 so that nothing is known about the cells that are generating the start-up values (which is which and what values are being generated).

If there are no correlations between the bits in different sequences, the probability of exactly *t* different bits appearing in *n* trials (HD = *t*) is given by a binomial distribution, as shown earlier in Equation (2). Given a large set of sequences, the distribution of Hamming distances obtained from all the possible comparisons between different pairs can be approximated by a normal distribution with expected value $n \cdot p$ and expected standard deviation $\sqrt{n \cdot p(1-p)}$, as stated in the de-Moivre-Laplace theorem.

If there are correlations between bits of different sequences,

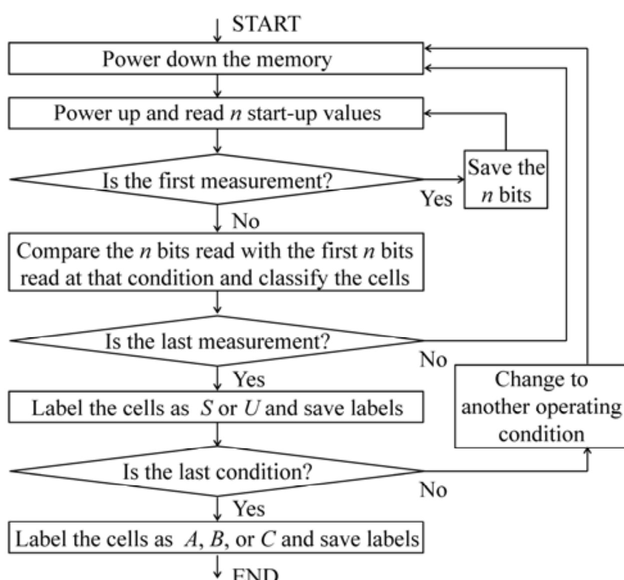

Fig. 1. Flowchart of memory cell classification.

Bernoulli trials remain binomially distributed but with a reduction in the number of independent trials or degrees of freedom, which becomes *N* instead of *n* (*N* < *n*) (see [28] for a more detailed explanation). Hence, given *k* sequences of *n* bits generated by SRAM start-up values, their independence is evaluated by the average inter fractional HD, and by the degrees of freedom, *N*, as follows:

$$\left[\frac{HD}{n}\right]_{\mu} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \quad (8)$$

$$N = \frac{\left[\frac{HD}{n}\right]_{\mu} \cdot \left(1 - \left[\frac{HD}{n}\right]_{\mu}\right)}{\left(\left[\frac{HD}{n}\right]_{\sigma}\right)^2} \quad (9)$$

where $\left[\frac{HD}{n}\right]_{\sigma}$ is the standard deviation of the inter fractional HD.

If the *n*-bit sequences are quite independent, $\left[\frac{HD}{n}\right]_{\mu}$ will be *p*= 0.5, $\left[\frac{HD}{n}\right]_{\sigma}$ will be $\sqrt{p(1-p)/n}$, and *N* will be *n*.

Another way to evaluate whether two *n*-bit sequences $R_i = X_{1i}, X_{2i}, \dots, X_{ni}$, and $R_j = X_{1j}, X_{2j}, \dots, X_{nj}$, where $X_{kl}=\pm 1$ ('0' values are converted to '-1'), are independent or non-correlated is to calculate their scalar product as:

$$correlation_{ij} = \sum_{k=1}^{n} X_{ki} \cdot X_{kj} = \sum_{k=1}^{n} c_{kij} \quad (10)$$

Correlation is zero (i.e., the sequences are independent) if the number of c_{kij} that are '1' is the same as the number of them that are '-1', producing a sum of zero. Thus, the condition for no correlation is that the sequence of c_{kij} must comprise independent Bernoulli random variables which take values of '1' or '-1' with the same probability of 0.5. This is equivalent to the condition of uniformity in the sequence of c_{kij}. The condition can be evaluated by the *NIST Frequency (Monobit) Test*, as commented below.

5) Uniqueness

The identifiers generated by genuine and fake memories should be sufficiently different to ensure the uniqueness of the identification. Uniqueness is achieved if the Hamming distance between IDs provided by genuine and fake devices (the interdie Hamming distance) is always greater than the Hamming distance between IDs provided by the genuine device (the intradie Hamming distance). Such condition also achieves the uniqueness of the secret key generated because it is possible to select the number of errors to be corrected by the ECC so that only the genuine device could be able to recover the response of the initialization step in the Code Offset-based Helper Data Algorithm.

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

If m identifiers generated by k devices are considered, an estimate of the uniqueness of the identification is given by the difference between the minimum interdie and the maximum intradie fractional Hamming distances, which is known as security distance (SD). This is defined as follows:

$$SD = \min_{\substack{x=1,\dots,k-1 \\ y=x,\dots,k \\ i,j=1,\dots,m}} \frac{HD(R_{ix}, R_{jy})}{n} - \max_{\substack{x=1,\dots,k \\ i=1,\dots,m-1 \\ j=i+1,\dots,m}} \frac{HD(R_{ix}, R_{jx})}{n} \quad (11)$$

As reliability increases, the maximum intradie Hamming distance decreases (tending to 0%) and, hence, the security distance increases. Similarly, as ID independence increases, the minimum interdie Hamming distance increases (tending to 50%), as does the security distance. In the ideal situation of 100% reliability and independence of identifiers, security distance will be 50%. A high security distance means that the probability of a genuine SRAM being rejected or a fake SRAM being accepted is quite small. In general, the security distance decreases as the length of the ID decreases (a detailed explanation of this can be found in [18]).

6) Unpredictability

Several tests can be applied to evaluate the unpredictability or randomness of a sequence. In this paper, tests taken from the NIST test suite for randomness [29] were applied, as described below. To be unpredictable, the n-bit sequence generated by n memory cells should be uniform (should have the same number of '1's and '0's). In other words, the fractional Hamming weight should be 0.5. This condition is tested by the *NIST Frequency (Monobit) Test*. Subsequences of an unpredictable sequence should also be unpredictable. The cumulative sums of the bits in the subsequences that can be formed from the complete sequence (considering '1' and '-1' values) should therefore also be zero, as in the complete sequence. This condition is tested by the *NIST Cumulative Sums Test*, in *forward* mode (if the subsequences are formed from the beginning to the end of the complete sequence) or in *backward* mode (if the subsequences are formed from the end to the beginning). The number of runs of '1's and '0's of various lengths should also be as expected for a random sequence, and the oscillation between zeros and ones should not be too fast or too slow. This is evaluated by the *NIST Runs Test*. For these tests, it is recommended that each sequence to be tested should have a minimum of 100 bits.

Each NIST test is statistical: a set of sequences, m, is tested and their p-values are obtained. The value of m should be in the order of the inverse of the statistical significance level. For example, for a significance level of α=0.01 (with a confidence, ρ, of 99% when determining unpredictability), about 1% of the sequences are expected to fail (p-value<0.01), 99% of the sequences are expected to pass the test (p-value≥0.01) and at least 100 sequences should be analyzed. NIST prescribes that the proportion of sequences that pass a statistical test should fall inside the confidence interval defined as:

$$\rho \pm 3\sqrt{\frac{\rho(1-\rho)}{m}} \quad (12)$$

NIST also examines the distribution of p-values to ensure uniformity and calculates a $Pvalue_T$ (a p-value of the p-values), which should - if the sequences are unpredictable - verify that:

$$Pvalue_T \geq 0.0001 \quad (13)$$

B. Evaluation strategy

Standard low-power dual-port 8-transistor TSMC 90-nm SRAM IP modules (TSDGA4096X60M8) were characterized experimentally. They were provided as IP modules by Europractice, with an operating voltage of 1.2V±10% and an operating temperature of -40ºC to 125ºC. The SRAM IP module was included in two different digital ASICs (hereafter referred to as ASICa and ASICb). The ASICs implement different signal processing algorithms, so their layouts are different (ASICa and ASICb are described in [30] and [31], respectively). In both cases, the SRAM module was used to store a set of parameters that the ASICs need for digital signal processing. The two ASICs therefore represented a real scenario in which to study the capability of an SRAM module to generate identifiers and true random numbers when powered-up, the SRAM itself forming part of a digital design.

The main variables considered in the SRAM operating conditions were the power supply voltage, Vdd, and temperature, T. Another critical factor is aging. The purpose of the evaluation strategy was to find a methodology capable of simultaneously increasing the reliability and entropy offered by a standard SRAM with respect to different operating conditions and aging, and which could be carried out at low cost and with the SRAM embedded in its operation context (because the behavior of the SRAM changes if the operation context changes - for example, if the power supply ramp-up time changes, as mentioned in the Introduction).

The first step taken to achieve that objective was to evaluate the percentage of stable and unstable cells (Equation (6) and Equation (7), respectively) obtained after considering several operating conditions, as illustrated in Figure 1. The number of measurements (start-ups) taken at each operating condition was 20. Ten integrated ASICa circuits were characterized. For each circuit, 2,280 bits (start-up values) provided by 40 x 57-bit words were registered. Those were enough bits to generate identifiers, secret keys and random sequences. Twenty integrated ASICb circuits were characterized. The SRAM module in ASICb (as happens with other ASICs) cannot be written/read from the input/output pins as easily as in ASICa. Hence, 168 bits (start-up values) provided by 14 x 12-bit words were registered for each circuit. Those were enough bits to generate identifiers.

Firstly, classification was performed taking into account only nominal operating conditions (Vdd=1.2V and T=25ºC). To evaluate the influence of power supply voltage in the

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

TABLE I
PERCENTAGES OF STABLE AND UNSTABLE CELLS, MAXIMUM INTRA HD AND MINIMUM ENTROPY

	Nominal conditions	With Vdd variations	With T variations	All conditions
Percentage of stable cells (ASICa)	91.86%±0.17%	89.70%±0.19%	82.58%±0.26%	78.56%±0.43%
Maximum intra HD (ASICa)	6.42%±0.12%	7.25%±0.13%	17.67%±0.23%	18.83%±0.25%
Percentage of stable cells (ASICb)	86.58%±0.62%	80.98%±0.84%	73.72%±0.65%	70.30%±0.77%
Maximum intra HD (ASICb)	8.36%±0.59%	12.97%±0.78%	20.43%±0.66%	22.70%±0.76%
Percentage of unstable cells (ASICa)	8.14%±0.17%	6.11%±0.10%	0.75%±0.04%	0.34%±0.04%
Minimum entropy (ASICa)	3.04%±0.09%	3.01%±0.08%	1.98%±0.08%	1.93%±0.07%

classification, the temperature was fixed to the nominal value and three operating conditions were considered, (1.08V, 25ºC), (1.2V, 25ºC), and (1.32V, 25ºC), covering the typical variations of ±10% of the nominal Vdd. To evaluate the influence of temperature in the classification, the power supply voltage was fixed to the nominal value and three operating conditions were considered, (1.2V, 5ºC), (1.2V, 25ºC), and (1.2V, 75ºC), covering variations above and below the nominal T. The ASICs were included on a printed circuit board (PCB) powered at 5V, but since the ASICa and ASICb cores (including the SRAM) needed 1.2V, a voltage regulator was used to generate the required voltage for the ASIC core. The regulator's output voltage was set by the ratio of two external resistors, one of which was a digital potentiometer. A digital switch made it possible to power down the ASIC (including the SRAM). A precision temperature forcing system, a Thermonics T-2650BV, was employed to control the operating temperature of the ASIC samples.

The second step was to evaluate the reliability (measured as $max_{IntraHD}$ in Equation (1)) and entropy (measured as H_{min} in Equation (5)) obtained without classifying the cells, that is, using stable and unstable cells to generate the responses. Firstly, responses generated at nominal operating conditions were analyzed. Secondly, responses generated at several operating conditions were also considered.

The third step was to evaluate how the classification and, subsequently, the adequate use of cells simultaneously improves the reliability and entropy under operating conditions not considered in the classification process. A detailed analysis was done with ASICa circuits to evaluate the improvements under nine operating conditions: (1) (1.08V, 25ºC), (2) (1.2V, 25ºC), (3) (1.32V, 25ºC), (4) (1.2V, 5ºC), (5) (1.2V, 75ºC), (6) (1.08V, 5ºC), (7) (1.32V, 5ºC), (8) (1.08V, 75ºC), and (9) (1.32V, 75ºC). The improvements in reliability were confirmed with more ASICb circuits under less operating conditions (the five conditions from (1) to (5), described above). The classification technique selected was that which offered the best trade-off between performance and cost.

The fourth step was to assess the advantages of using the selected technique to generate identifiers, secret keys, and true random numbers, testing particularly the independence and unpredictability of the generated sequences under nominal operating conditions.

Finally, the fifth step was to evaluate how the selected classification technique responds to aging. Three integrated ASICa circuits were characterized. For each circuit, 91,200 bits (start-up values) provided by 4 x 400 x 57-bit words were registered. One of the ASICa circuits was used to test how the stable and unstable cells change in time under nominal operating conditions and normal SRAM activity. The other circuits were used to test the evolution under operating conditions that accelerate aging and/or NBTI (Negative Bias Temperature Instability), one of them working continuously at increased power supply voltage (1.32V, 25ºC) and the other at increased temperature (1.2V, 75ºC). A climatic chamber ACS-EOS 200TC was used to carry out the last experiments.

The inputs to configure the ASICs were provided by an Agilent 16720A Pattern Generator while the outputs (including the start-up values of the SRAM) were recorded by an Agilent 16823 Logic Analyzer. To avoid problems of data remanence, the tested ASICs were kept shut down for a fair amount of time (30 seconds) between start-ups, as suggested in [32]. A Matlab program running on a computer was used to automatize the characterization measurements with the Instrument Control Toolbox. The Matlab software reported in [33] was adapted for this purpose.

III. WAYS TO IMPROVE RELIABILITY AND ENTROPY

A. *Influence of stable and unstable cells*

The percentages of stable and unstable cells, as computed in Equation (6) and Equation (7), respectively, were measured in 10 samples from ASICa (2,280 bits per sample) and 20 samples from ASICb (168 bits per sample), considering 20 measurements per operation condition analyzed, following the flowchart in Figure 1.

The mean values of the percentage of stable cells, together with their standard errors, are shown in the first and third rows of Table I. The values in the first column considered only nominal operating conditions (1.2V, 25ºC). The values in the second column considered three operating conditions with three different Vdd values at nominal temperature, (1.08V, 25ºC), (1.2V, 25ºC), and (1.32V, 25ºC). The values in the third column considered three operating conditions with three different T values at nominal Vdd (1.2V, 5ºC), (1.2V, 25ºC), and (1.2V, 75ºC). Finally, the values in the fourth column considered all the operating conditions analyzed (nine for ASICa and five for ASICb). From the results, it can be concluded that the cells that are always stable (*A* cells) decrease if Vdd variations are considered and decrease even more if T variations are considered.

The maximum intra HD, as described in Equation (1), was calculated for the ASICa and ASICb samples: (a) under

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

nominal conditions, (b) with three different Vdd values at nominal temperature, (c) with three different T values at nominal Vdd, and (d) considering all the conditions analyzed. The mean values for ASICa (obtained after evaluating 10 samples from ASICa, with 13 sets of 128 bits per sample) are shown together with their standard errors in the second row of Table I. The pairwise comparisons calculated per sample and per set of 128 memory cells were: (a) 3,160 pairs among 80 responses under nominal conditions (the number of responses is m in Equation (1)), (b) and (c) 28,680 pairs among 240 responses, and (d) 258,840 pairs among 720 responses under all conditions. Similarly, the maximum intra HD was calculated for 20 samples from ASICb and 1 set of 128 bits per sample. The mean values with their standard errors are shown in the fourth row of Table I. The pairwise comparisons calculated per sample and set of 128 memory cells were: (a) 190 pairs among 20 responses at nominal conditions (m=20 in Equation (1)), (b) and (c) 1,770 pairs among 60 responses, and (d) 4,950 pairs among 100 responses at all conditions. In both ASICs, it can be seen how the maximum intra HD increases as the percentage of stable cells decreases.

The mean values of the percentage of unstable cells, with their standard errors, are shown in the fifth row of Table I. As occurred with the stable cells, the cells that are always unstable (*B* cells) were found to decrease with Vdd variations and decrease even more with T variations.

Minimum entropy, as described in Equation (5), was calculated for the 2,280 memory cells analyzed in the 10 samples from ASICa. Figure 2 illustrates how the minimum entropy of one of the samples under nominal operating conditions changes versus the number of measurements considered to evaluate the p_{imax}. With 100 start-up measurements, H_{min} converges to its asymptotic value. 100 start-ups were therefore considered to evaluate the maximum probability observed for each memory cell (p_{imax}). The H_{min} was calculated: (a) under nominal conditions, (b) with three different Vdd values at nominal temperature, (c) with three different T values at nominal Vdd, and (d) considering all nine conditions. The minimum value was recorded when several

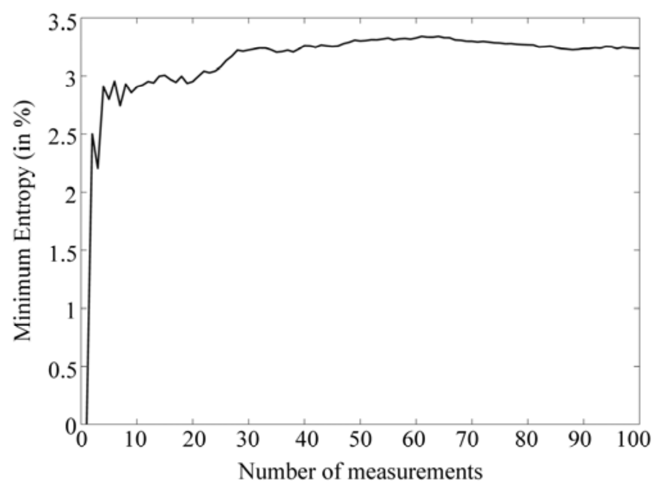

Fig. 2. Convergence of the minimum entropy to its asymptotic value.

operating conditions were considered. The mean values with their standard errors are shown in the sixth row of Table I. It can be seen how the H_{min} is low because the percentage of unstable cells is low and that it decreases as the percentage of unstable cells decreases.

The next analysis was aimed at evaluating how the classification of cells into stable and unstable cells can simultaneously improve both the reliability and randomness of the SRAM. The analysis was carried out with ASICa samples and confirmed with ASICb samples. To classify the cells, several operating conditions can be analyzed: the more conditions are analyzed, the more accurate the classification. However, it is impractical to take into account too many conditions and, in the case of unstable cells, as shown in the fifth column of Table I, the number of unstable cells may be too small. Two forms of classifying the cells as stable and unstable were therefore analyzed: (a) taking into account the Vdd values and considering the three operating conditions (1.08V, 25ºC), (1.2V, 25ºC), and (1.32V, 25ºC), and (b) taking into account the T values and considering the three operating conditions (1.2V, 5ºC), (1.2V, 25ºC), and (1.2V, 75ºC). In both cases, classification was carried out by considering 20 measurements per operation condition.

B. *Reliability improvement with classification*

To evaluate the reliability improvement, the start-up values provided by the SRAM cells were organized as 13 sets of 128 bits per each ASICa sample and evaluated per each operation condition (a maximum of 13 sets per sample can be generated by the *A* cells found with T variations in all the samples). The maximum intra HD between 80 responses generated by the same 128 cells at different times was calculated for three situations: (a) with no classification, so that the 128 cells were any of the memory cells in the SRAM; (b) with classification taking into account Vdd and (c) with classification taking into account T values, so that the responses were generated by 128 memory cells classified as *A* cells.

Table II shows the mean values (obtained after averaging over the 10 samples from ASICa and the 13 sets per sample) and the standard errors of the maximum intra HD under each operation condition. It can be seen how using stable cells always improves reliability. Classification taking into account Vdd values provides the best reliabilities under operating

TABLE II
MEAN AND STANDARD ERRORS FOR $MAX_{INTRAHD}$ IN ASICA SAMPLES UNDER EACH OPERATION CONDITION (*A* CELLS ARE USED WITH CLASSIFICATION)

Vdd (V), T(ºC)	Without classification	With classification based on Vdd	With classification based on T
(1.08, 25)	6.41%±0.13%	0.99%±0.06%	1.17%±0.07%
(1.2, 25)	6.42%±0.12%	0.87%±0.06%	1.12%±0.07%
(1.32, 25)	6.56%±0.12%	1.02%±0.06%	1.13%±0.07%
(1.08, 5)	7.45%±0.13%	4.45%±0.14%	1.94%±0.08%
(1.2, 5)	7.85%±0.20%	4.47%±0.13%	1.97%±0.15%
(1.32, 5)	7.62%±0.13%	4.78%±0.14%	2.19%±0.08%
(1.08, 75)	4.81%±0.14%	2.03%±0.11%	1.02%±0.09%
(1.2, 75)	4.65%±0.12%	1.94%±0.10%	0.76%±0.06%
(1.32, 75)	4.75%±0.12%	2.10%±0.10%	1.03%±0.08%

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

TABLE III
MEAN AND STANDARD ERRORS FOR $MAX_{INTRAHD}$ IN $ASIC_A$ AND ASICB SAMPLES UNDER ANY OPERATION CONDITION (*A* CELLS ARE USED WITH CLASSIFICATION)

	Without classification	With classification based on Vdd	With classification based on T
ASICa	18.83%±0.25%	6.80%±0.25%	5.76%±0.26%
ASICb	23.30%±0.80%	10.89%±0.57%	4.87%±0.56%

conditions with nominal temperature and also improves reliability under the other operating conditions. Classification taking into account T values provides the best reliabilities under operating conditions with non-nominal temperature and also improves reliability under the other operating conditions.

To test the statistical significance of these results, a two-sample Kolmogorov-Smirnov test was performed to compare (a) the distributions of the maximum intra HD obtained without classification and with Vdd classification, and (b) the distributions of the maximum intra HD obtained without classification and with T classification. This nonparametric test was applied because the underlying distribution of the maximum intra HD is unknown. The null hypothesis was that the maximum intra HDs calculated with and without classification come from the same continuous distribution (i.e., classification is not relevant). The alternative hypothesis was that they come from different distributions (i.e., the influence of classification is relevant). If the p-value resulting from the test is low, it means that the two distributions are different, but if the p-value is close to 1, it means that the two distributions are similar. The p-value is very accurate for large sample sizes, and reasonably accurate for sample sizes n_1 and n_2, such that $(n_1 \cdot n_2)/(n_1 + n_2) \geq 4$. In these tests, $n_1 = n_2 = 130$, so the p-values obtained were reasonably accurate. In both cases, the test rejected the null hypothesis at the 5% significance level. The p-value obtained in test (a) was 5.7e-51 (and the maximum difference between the empirical cumulative distribution functions was 0.93). The p-value obtained in test (b) was 4.1e-56 (and the maximum difference between the empirical cumulative distribution functions was 0.98). The conclusion is that classification of the cells modifies the distribution of the maximum intra HD. The advantage is that the maximum intra HD becomes smaller, as is desirable for ID and secret key generation.

The real scenario is that IDs or secret keys can be registered under one set of operating conditions and should then be verified under different operating conditions (not under the same conditions, as calculated in the results of Table II). If the maximum intra HD between responses provided by the same memory cells under all the possible operating conditions is calculated, the mean values and standard errors obtained for all the samples from ASICa and the 13 sets considered per sample are shown in the first row of Table III. In the second row of Table III similar behavior can be observed in the 20 samples from ASICb with one set per sample (with 122 bits because a maximum of 122 *A* cells was found with T variations in all the samples).

TABLE IV
MEAN AND STANDARD ERRORS FOR H_{MIN} IN $ASIC_A$ SAMPLES UNDER EACH OPERATION CONDITION (*B* CELLS ARE USED WITH CLASSIFICATION)

Vdd (V), T(ºC)	Without classification	With classification based on Vdd	With classification based on T
(1.08, 25)	3.16%±0.06%	42.21%±0.77%	39.06%±1.48%
(1.2, 25)	3.04%±0.09%	42.52%±1.06%	40.67%±2.06%
(1.32, 25)	3.08%±0.07%	42.33%±0.91%	41.60%±1.41%
(1.08, 5)	3.65%±0.07%	10.39%±0.82%	29.32%±3.33%
(1.2, 5)	3.64%±0.08%	10.54%±0.58%	31.24%±3.14%
(1.32, 5)	3.73%±0.07%	10.58%±0.79%	30.07%±3.56%
(1.08, 75)	2.13%±0.11%	14.56%±0.96%	29.98%±3.16%
(1.2, 75)	1.98%±0.08%	13.80%±0.60%	30.57%±2.33%
(1.32, 75)	2.04%±0.06%	14.02%±0.56%	28.89%±3.06%

C. Entropy improvement with classification

Minimum entropy was calculated using Equation (5) for three situations: (a) with no classification of the cells, therefore considering the 2,280 bits analyzed, (b) considering only the cells classified as unstable under the three operating conditions with nominal T and varying Vdd, and (c) considering only the cells classified as unstable under the three operating conditions with nominal Vdd and varying T. Table IV shows the mean values (the average of 10 samples from ASICa) and the standard errors of the minimum entropy under each operation condition. It can be seen how classification taking into account Vdd variations provides the highest minimum entropy under operating conditions with nominal T, and also considerably improves entropy under other operating conditions. Classification taking into account T variations provides the highest minimum entropy under operating conditions with non-nominal T. The problem is that the number of unstable cells found with T variations was very small (0.75% on average, as shown in Table I), which is impractical for many applications (it represents an average of 17 cells in 2,280). On the other hand, the number of cells classified as unstable with Vdd variations was 6.11% on average (an average of 139 cells in 2,280).

A two-sample Kolmogorov-Smirnov test was performed to test the statistical significance of the minimum entropy results. The distribution of the minimum entropy obtained with 10 samples from ASICa under the nine operating conditions (90 values) without classification was compared with (a) the distribution with Vdd-based classification and (b) the distribution with T-based classification. With such sample sizes (90 data), the p-values obtained were reasonably accurate. Again, this nonparametric test was applied because the underlying distribution of the minimum entropy is unknown. The null hypothesis was that the minimum entropies calculated with and without classification come from the same continuous distribution (i.e., classification is not relevant). The alternative hypothesis was that they come from different distributions (i.e., the influence of classification is relevant). In both cases, the test rejected the null hypothesis at the 5% significance level. The p-value obtained for both tests was 4.1e-41 (and the maximum difference between the empirical

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

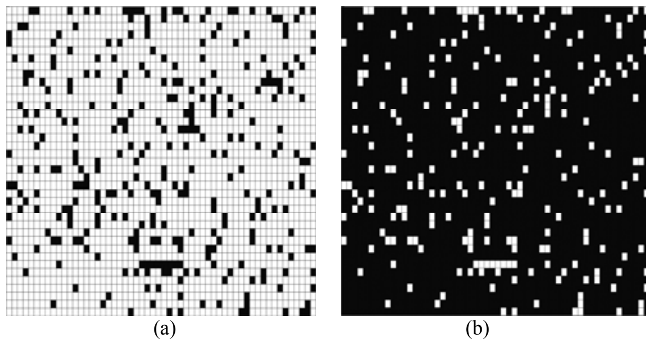

Fig. 3. (a) Stable cells depicted in white and (b) unstable cells depicted in white after classification based on 20 measurements per Vdd value.

cumulative distribution functions was 1). The conclusion is that classification of the cells modifies the distribution of the minimum entropy. The advantage is that the minimum entropy becomes bigger, as is desirable for random number generation.

IV. Classification Based on VDD Variations

Classification taking into account Vdd variations provides a good trade-off between improvements in reliability and entropy and simplicity of realization. Furthermore, the number of unstable cells detected is not as small as when taking into account T variations. This was therefore the classification method selected to be implemented and studied in greater detail. The way the method was applied and the advantages obtained in the ASICa and ASICb samples are summarized below.

No spatial correlation was detected in the stable and unstable cells found by the classification. Figure 3 shows the 40 x 57 memory cells analyzed in one of the ASICa samples. The stable cells (*A* cells) can be seen in white in Figure 3a and the unstable cells (*B* cells) can be seen in white in Figure 3b In all the samples, there are no rows or columns in the SRAM with the same *A* cells detected. In the case of *B* cells, since their number is much smaller, there are no rows in all the SRAMs with the same B cells, but some of the columns are equal (particularly, those with one or no *B* cells). These columns are different in the different ASICa samples.

A. Advantages for the generation of identifiers

1) Independence of the IDs

The start-up values provided by the *A* cells of the same SRAM should be independent to ensure that an ID cannot be ascertained through knowledge of other IDs provided by the same SRAM. At least 1,800 *A* cells were detected in four samples from ASICa with the proposed classification. They were organized as 18 x 100-bit IDs. To evaluate the independence of the IDs, the fractional HDs between all the pairwise combinations of the 18 IDs (generated under nominal conditions) were calculated, giving a total of 153 HDs. Rather than only one set, 100 sets of 18 x 100-bit IDs were analyzed in order to consider all the possible displacements between the IDs generated: i.e., the first set considers that the first *A* cell of the first ID is the first *A* cell found in the SRAM; the second set considers that the first *A* cell of the first ID is the second *A* cell found in the SRAM (thus, the 18 IDs generated are displaced by 1 bit in relation to the first set); and so on, till the 100th set, which considers that the first *A* cell of the first ID is the 99th *A* cell found in the SRAM. A total of 15,300 HDs were therefore calculated per ASIC sample. The objective was to discover any possible correlation in the IDs generated. Figure 4 shows the distribution of 61,200 fractional HDs obtained from four ASICa samples, and how the histogram can be approximated by a normal distribution, in this case with a mean value of $\left[\frac{HD}{n}\right]_{\mu}$=0.502 and a standard deviation of $\left[\frac{HD}{n}\right]_{\sigma}$=0.052. Applying Equation (9), the number of independent bits in the sequences is 92.59%, which means a very high level of independence among the IDs provided by the same SRAM, as summarized in Section II.A.4.

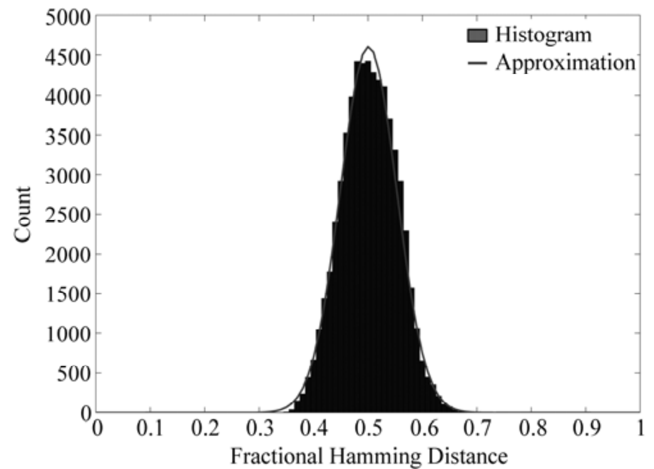

Fig. 4. Distribution of fractional HDs between IDs from the same SRAM.

To further analyze correlation, the scalar product between the 153 pairs of IDs per sample was calculated as in Equation (10). The uniformity of the 153 sequences of 100 c_{kij} correlation bits was analyzed using the *NIST Frequency Test*. The proportions of sequences with a p-value bigger than 0.01 (a confidence of 99%) verify the condition in Equation (12) and the distributions of p-values verify Equation (13). It can therefore be concluded, with a confidence of 99%, that the IDs provided by the same SRAM are not correlated.

IDs generated by different SRAMs should also be fairly independent to ensure their uniqueness. That is to say, an ID should be generated by only one single ASIC sample. To test this feature, the interdie fractional Hamming distances of 128-bit IDs obtained from the 20 ASICb samples after applying classification were calculated (95,000 HDs were calculated considering all operating conditions). Again, the histogram can be approximated by a normal distribution, in this case with a mean value of $\left[\frac{HD}{n}\right]_{\mu}$=0.483 (which is very close to 0.5, as is desirable for independent IDs) and a standard deviation of $\left[\frac{HD}{n}\right]_{\sigma}$=0.0447. Applying Equation (9), the number of

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

TABLE V
RESULTS REPORTED BY NIST TESTS APPLIED TO 72 IDS OF 100 BITS FROM ASICA SAMPLES

Test	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	$Pvalue_T$	Proportion
Frequency	8	7	6	7	10	11	10	0	8	5	0.1695	0.92
Cumulative sums (fw)	11	6	10	8	3	7	3	2	17	5	0.0015	0.94
Cumulative sums (bw)	7	7	7	9	6	5	5	6	16	4	0.0949	0.96
Runs	14	9	8	7	2	6	6	2	7	11	0.0424	0.89

independent bits in the sequences is 97.64%. Hence, the comparison between each pair of bits in the IDs generated by different SRAMs (as also happens with the IDs generated by the same SRAM) can be approximated by a Bernoulli trial as explained in Section II.A.4.

2) Unpredictability of the IDs

To evaluate the unpredictability of the IDs generated by the *A* cells both from the same and from different SRAMs, the *NIST Frequency Test*, *Cumulative Sums Test* (*forward* and *backward*) and *Runs Test* were applied to the 18 x 100-bit IDs obtained from the ASICa samples. The IDs from four samples were analyzed (a total of 72 x 100-bit sequences). A confidence of 95% was selected for the tests on this set of sequences. The results for the uniformity of p-values and the proportion of passing sequences are shown in Table V. Columns C1 to C10 correspond to the frequencies of p-values in the ten discrete bins into which the interval of values can be divided. Column P_{valueT} is the p-value of the p-values and the last column shows the proportion of sequences with a p-value bigger than 0.05. All the tests were passed (Equations (12) and (13) were fulfilled), so the IDs can be said to be unpredictable with a confidence of 95%. In particular, the *Frequency Test* ensured that the numbers of 0's and 1's in the IDs were distributed uniformly.

3) Uniqueness of the IDs

Classification based on Vdd variations allows generating IDs with much smaller lengths, but the same separation between genuine and fake populations, because the maximum intradie Hamming distance is much smaller and the minimum interdie Hamming distance does not vary (or even increases slightly). For example, the security distances (calculated as in Equation (11)) obtained with 20 samples from ASICb are shown in Table VI for different ID lengths. It can be seen how 70-bit IDs provide the same security distance than 128-bit IDs without classification. This represents a reduction of 45% in the ID length with the same security.

As the length of the identifiers shortens thanks to classification, the number of different IDs that can be generated by a SRAM (known as the number of challenge-response pairs in a SRAM PUF) increases. For example, in the SRAM analyzed in both ASICs, more than 80% on average of the memory cells are stable with Vdd variations (cells labeled as A) and the responses can be reduced in length by around 45%. The new number of challenge-response pairs, CRP_{new}, with a similar security distance to the original number, CRP_{orig}, is therefore:

$$CRP_{new} = \frac{0.8 \cdot SRAM_{size}}{0.55 \cdot ID_{orig}} = 1.45 \cdot CRP_{orig} \quad (14)$$

Hence, the number of different IDs that can be generated from the SRAM analyzed increases by a factor of 1.45 thanks to classification.

B. Advantages for the generation of secret keys

Several ECCs have been reported in literature for extracting secret keys from PUFs. The codes should correct random rather than burst errors. Hence, neither Reed-Solomon nor convolutional codes, which have very good burst error correcting capabilities, are adequate for PUFs. LDPC codes are very efficient but require very large, sparse binary matrices, making them inadequate for hardware implementations or platforms with low computational resources. BCH codes have a very good error correcting capability, but their decoders are very complex to be efficient in hardware. As analyzed in [26], the simplest codes for implementation (particularly in hardware) are repetition codes and Reed-Muller codes of the first order. We therefore focused on these two types of ECCs to correct the errors produced in the SRAM start-up values.

An $[n,k,d]$-code means a binary code C with codeword length n, message length k, and minimum distance d. Repetition codes have d=n and can correct up to $t = \left\lfloor \frac{n}{2} \right\rfloor$ errors. Reed-Muller codes of the first order, usually denoted as $R(1,m)$, are linear codes with parameters $n = 2^m$, $k = m+1$, and $d = 2^{m-1}$. They have the capability to correct up to $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

As summarized in Section II.A.1, if the worst case bit error probability, p, is known (or estimated by the maximum intra HD) and the ECC is able to correct up to t errors, then the probability of a sequence having more than t errors is given by P_{total} in Equation (3). The authors in [13] and [26] consider that a P_{total} of 10^{-6} is a conservative value that fulfills the requirements of most of typical security applications.

The performance of Reed-Muller and repetition codes is illustrated in the ASICa samples. As was shown in Table III,

TABLE VI
SECURITY DISTANCES FOR DIFFERENT ID LENGTHS IN ASICB SAMPLES

	Without	With classification based on Vdd				
ID bits	128	60	65	70	80	105
Security Dist. (in %)	17.40	16.05	16.37	17.67	19.47	23.03

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

TABLE VII
RESULTS OBTAINED WITH DIFFERENT ERROR CORRECTING CODES

Without classification			With classifications based on Vdd		
ECC	P_{TOTAL}	Source bits	ECC	P_{TOTAL}	Source bits
R(1,9)=RM[512,10,256]	3.38e-4	6,656	R(1,6)=RM[64,7,32]	4.36e-6	1,216
Repetition[39,1,39]	5.17e-6	4,992	Repetition[13,1,13]	7.99e-6	1,664

TABLE VIII
RESULTS REPORTED BY NIST TESTS APPLIED TO 100 START-UP SEQUENCES OF 100 MEMORY CELLS

Test	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	$Pvalue_T$	Proportion
Frequency	9	6	8	14	8	19	14	0	11	11	0.0043	0.96
Cumulative sums (fw)	11	2	8	13	8	9	8	9	16	16	0.0669	0.98
Cumulative sums (bw)	10	4	7	10	6	9	8	15	20	11	0.0235	0.98
Runs	8	11	10	9	11	10	15	7	9	10	0.8978	0.99

the worst case bit error probability if classification is not applied is 18.83% on average. Using this probability, the left part of Table VII illustrates the P_{total} achieved by the Reed-Muller and repetition ECCs. Even a Reed-Muller code $R(1,9)$ is not able to fulfill a P_{total} in the order of 10^{-6}. A repetition code would need a codeword length of 39 bits to reach that order of P_{total}. As summarized in Section II.A.1, the responses generated by the SRAM should be of the same length as the codewords, n, (or a multiple of n) to calculate the helper data in the Code Offset-based Helper Data Algorithms. If a secret of s error-free bits is required, $n \cdot \left\lceil \frac{s}{k} \right\rceil$ bits should be provided by the SRAM (these bits will be named as source bits for the $[n,k,d]$-code). The left part of Table VII illustrates the source bits (without applying classification) required by each ECC to obtain a secret of 128 error-free bits. In both cases, the number of source bits is very high.

With classification based on Vdd variations, the worst case bit error probability is reduced to 6.80% on average for ASICa. In this case, the right part of Table VII shows that a Reed- Muller code $R(1,6)$ and a repetition code Repetition[13,1,13] would be able to fulfill a P_{total} in the order of 10^{-6}. The number of source bits to be processed is reduced by 82% in the case of Reed-Muller codes and by 67% in the case of repetition codes. Classification reduces not only the number of bits to be processed but also the complexity of the two ECCs (the Reed-Muller and the repetition ECCs). For example, if the low-resource first-order Reed-Muller decoder reported in [26] is employed, which requires $(2^m + 6m - 1)$ Flip Flops, $(\frac{m^2}{2} + \frac{13m}{2} - 2)$ XOR gates, 9m AND gates, $(\frac{5m^2}{2} + \frac{m}{2} - 1)$ OR gates, and m NOT gates, sequential elements are reduced by 82% and combinatorial elements by 47% since m is reduced from 9 to 6. In the case of repetition decoders, which can be implemented by a counter and a comparator, a 4-bit counter instead of a 6-bit counter is enough.

The unpredictability of the responses provided by the SRAM cells (described in the subsection above) ensures that no information is leaked by the helper data stored for secret key generation, as is desirable.

C. Advantages for the generation of random numbers

The bit strings generated from the start-up values of a given set of SRAM cells are not very random. If around 80% of the memory cells are stable, that means that 80% of the bits in the strings are always the same: i.e., the probability of those cells producing a “0” or “1” is approximately 1. As a matter of fact, in the SRAM analyzed, the minimum entropy value when no classification was applied ranged from 2.67% to 3.46% under nominal operating conditions. The unpredictability of start-up values under nominal conditions was analyzed for the ASICa samples by applying the *Frequency*, *Cumulative Sums* and *Runs Tests* from the NIST Test Suite, as explained in Section II.A.5. 100 start-up sequences provided by 100 memory cells were analyzed and a confidence of 99% was selected. If no classification was applied, the sequences failed all the tests.

With classification based on Vdd variations, minimum entropy increased to between 37.28% and 47.15% under nominal operating conditions. Considering again 100 start-up measurements of 100 bits, but now generated by 100 memory cells classified as B in the registration phase, all the above mentioned NIST tests were passed (Equations (12) and (13) were fulfilled). Table VIII shows the results for the uniformity of the p-values and the proportion of passing sequences provided in the final analysis report after completing the tests on one of the ASICa samples. The last column shows the proportion of sequences with a p-value bigger than 0.01. It can therefore be concluded with a confidence of 99% that the sequences are random.

If better performance is required of sequences generated from the SRAM start-up values, post processing functions must be implemented to eliminate bias and increase entropy (i.e., the entropy distillation process). The most widespread methods used for this purpose are the XOR combination [34], Von Neumann’s corrector, and hash functions [5], [23]. The advantages of using the proposed method will be illustrated using the latter approach, which, for SRAMs, was adopted in [5], [23]. The minimum entropy value defines the amount of source bits that must be processed by the hash functions in order to obtain a full entropy bit string. In the SRAM analyzed, the minimum entropy value under the worst

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

TABLE IX
ADVANTAGES TO OBTAIN 256-BIT STRINGS WITH FULL ENTROPY

Source bits needed		Time using the SHA-256 in [23]	
Without	With classification	Without	With classification
1,616 bytes	308 bytes	8.08 µs	1.54 µs

operating conditions when the method was not applied was 1.98% on average (as shown in Table IV). 808 bytes are therefore necessary to obtain a fully random seed of 128 bits, 1616 bytes to obtain one of 256 bits, and so on. If the classification is done based on Vdd variations, the minimum entropy value under the worst operating conditions increases to 10.39% on average (as shown in Table IV). This means that 154 bytes are required for a fully random seed of 128 bits, 308 bytes for one of 256 bits, and so on. This represents a reduction of 81% in the input data that hashes must process. For example, if the lightweight SHA-256 structure described in [23] is used, the time required to obtain a fully random sequence is calculated as (*source_bits* · *hash_cycles*) / (*hash_input_bits* · *frequency*). Since the input block size of the structure (*hash_input_bits*) is 512 bits and the hash is executed in 64 clock cycles (*hash_cycles*), the time required for generating the random numbers at a clock frequency of 200MHz (*frequency*) is reduced, as is summarized in Table IX. The reduction in processing time is even more noticeable if lightweight hash realizations with lower throughput are employed.

D. Advantages for implementation

For use in a real scenario, the classification method based on Vdd variations involves two phases. In the first phase (hereafter referred to as the registration phase), the cells of the SRAM analyzed are labeled as *A* and *B*. In the second phase (hereafter referred to as the generation phase), the start-up values of the cells analyzed in the registration phase are read at nominal Vdd, so as to generate one bit string with the start-up values of *A* cells (for IDs and secret keys) and another with the start-up values of *B* cells (for random numbers). According to the experimental results discussed above, a good registration phase is usually ensured if the range in which the memory can operate normally is analyzed, which is usually ±10% of the nominal value, V_{ddnom} (for example, the three values $0.9 \cdot V_{ddnom}$, V_{ddnom}, and $1.1 \cdot V_{ddnom}$, are all adequate). The method can therefore be applied without unplugging the memory from its operation context. Also, the operating temperature does not need to be controlled, eliminating the need for big, costly temperature forcing systems.

The first phase should be carried out at least once to classify the cells. Once the cells are classified, the identifiers and random numbers are generated by executing the second phase, with no need to repeat the first phase. The first phase can be repeated, if necessary, during the lifetime of the SRAM to refine the classification, considering other possible field operating conditions (in particular, other ambient temperatures). In this case, the cells labeled as *A* (or *B*) are the cells again labeled as *A* (or *B*). The rest of the cells are labeled as *C*. Care should be taken because, as has been shown in the previous section, the number of *B* cells may be very low if several operating conditions are considered (the number of *A* cells is usually high). With this in mind, the number of cells could be counted at the registration phase to ensure a minimum number of the required cells at all times. It is important to have a minimum number of *A* cells to generate IDs and/or secret keys and a minimum number of *B* cells to generate random numbers. When the first phase is carried out without indicating that it has been repeated, all the previously stored labels are deleted and the newly obtained labels are stored. In the latter case, the registration phase can adapt the classification to take into account SRAM aging, as commented below.

The registration phase should always be carried out using an appropriately authorized verification system (known as a Trusted Third Party or TTP). If the registration phase is repeated, the previous IDs or keys are revoked and the new ones are registered by the Trusted Third Party. Taking into account the large number of attacks that may compromise an ID or, especially, a key, the capability to allow the easy creation of new registrations is another advantage of the proposed method.

The number of measurements, *Q*, taken at the same power supply, Vdd, to carry out the registration phase should not be too small or too large. As an example, Figure 5a illustrates the number of stable cells found versus the *Q* measurements taken at each of the three Vdd values for one of the ASICa samples. A small number of measurements does not detect stable cells well. As the number of measurements increases, the number of cells classified as *A* decreases, although there is a number above which the rate of decrease becomes very small. On the other hand, a large number of measurements fails to detect adequate unstable cells. As the number of measurements

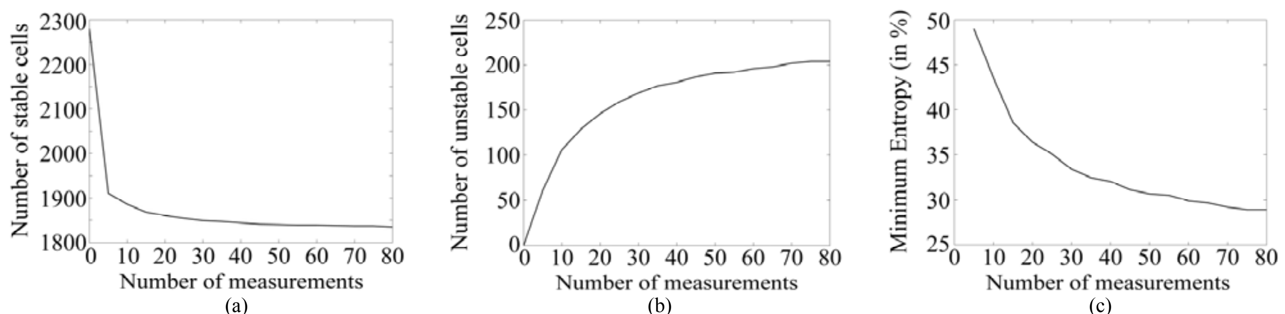

Fig. 5. Number of (a) stable and (b) unstable cells, and (c) minimum entropy at nominal conditions versus the number of measurements.

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

increases, the number of cells classified as *B* increases. This is illustrated in Figure 5b for the same ASICa sample, with the *Q* measurements taken at each of the three Vdd values. As the number of measurements increases, the probability of bit flipping decreases (a cell is classified as unstable if its start-up value changes only once), which means that the minimum entropy provided by the cells classified as *B* decreases. This is illustrated in Figure 5c, which shows the minimum entropy provided by the unstable cells selected at each number *Q* of measurements for the same ASICa sample, calculated under nominal conditions. In the SRAMs analyzed, *Q*=20 measurements were selected as a good trade-off for classification (this is why the results shown in the previous sections considered 20 start-ups per operating condition when carrying out classification).

The blocks needed to implement the classification method are the following. (a) A *voltage block* to power down the memory or power it up to the adequate power supply voltage. (b) A *classification block* to analyze which cells of the memory should be labeled as *A* and which as *B*, if the operation mode is the registration phase. If the operation mode is the generation phase, this block sends the classification results to the control block. (c) A *control block* that is in charge of all the other blocks and indicates when the method should be initialized, which phase is to be executed, whether or not the first phase is to be repeated, the lengths of the strings to be generated, the *P* values of the power supply voltages to be analyzed and the *Q* measurements to be taken per Vdd. The control block tells the voltage block when the memory should be powered up or down and, in the former case, the value of the Vdd. It also enables the reading of the memory, by activating the corresponding signals, tells the classification block the current operation mode, and, if the operation mode is the generation phase, it employs the labels stored in the registration phase to generate identifiers and/or random numbers.

The time required for the generation phase depends on the time needed to read the memory cells (which is in the range of nano or microseconds depending on the memory). The time required for the registration phase, $time_{reg}$, is determined by the values P, Q, and the power-down time, $time_{down}$, so that $time_{reg} \cong time_{down} \cdot Q \cdot P$. The $time_{down}$ values required to avoid data remanence can range from several microseconds to several seconds, depending on the memory and its operation context [11], [32]. In any case, the registration phase can be carried out when the memory is not employed in its application context, since the memory is idle for long intervals in many applications. For example, many embedded systems for sensing applications such as thermostats, keyless entry, security systems, etc., employ microcontrollers and SRAMs that are active only during short bursts of time (often, only milliseconds) and are idle during intervals of many tens of seconds. Such microcontrollers are therefore already equipped with a voltage block that allows a deep sleep mode to reduce power consumption, in which the entire microcontroller, including the on-chip SRAM, is powered down [35]. Hence, another advantage of the proposed method is that the voltage block is already available in many systems and, if it is included, it can be exploited also to power down the SRAM when it is not being used by the digital signal processing, thus saving power.

The operations carried out by the control and classification blocks to implement the proposed classification are very simple. A VLSI module was therefore designed which can be incorporated into the digital ASIC where the SRAM is embedded. The register transfer level (RTL) specification of the module was written in the VHDL hardware description language. A synthesis tool (Design Analyzer from Synopsys) transformed the RTL specification into a set of logic gates, considering TSMC 90-nm technology information. The place and route tool employed was System-on- Chip (SoC) Encounter from Cadence. The active area of the module is 42,025 μm^2. The active area of the 8T TSMC 90-nm SRAM IP modules (TSDGA4096X60M8) analyzed is 677,543.051 μm^2. The designed module therefore represents an area overhead for the SRAM of 6.20% compared to an SRAM without this additional module. In ASICa and ASICb, both of which have an active area of 3.46 mm^2, the designed module occupies 1.21% of the active area.

E. Advantages for countering aging

Aging phenomena affect integrated circuit behavior over time. In the case of SRAMs the dominant factor is the Bias Temperature Instability, BTI, and more specifically (as its impact is higher than the PBTI), the Negative Bias Temperature Instability, or NBTI. An overview of NBTI and other aging effects can be seen in [36]. NBTI may increase the theshold voltage (VTH) of the PMOS transistors subjected to negative gate to source bias due to high temperature stress conditions [37]-[38]. If the preferred start-up value of a cell is stored during a long time, the PMOS transistor with smaller VTH is turned on and its VTH may increase because of NBTI degradation. Hence, that start-up value may become less preferred and even the opposite value may become preferred. The consequence is that memory cells can be made more unstable. On the contrary, if the opposite of the preferred startup value is stored during a long time, the cells can be made more stable [11], [39].

To evaluate aging phenomena, three samples of ASICa circuits were considered. First, the classification method based on Vdd variations was applied as commented above (three Vdd values and 20 measurements per Vdd operating condition) to 4 x 10 blocks of 2,280 bits (91,200 bits) per sample. The stable (*A*) and unstable (*B*) cells were identified per each block. After, one of the ASICs was working in normal mode (acting as a non-linear controller for automotive applications), under nominal operating conditions (1.2V, 25ºC) during 4 weeks (the circuit was controlling during 8 hours per work day and powered down the rest of the day, except for the time of the registration phase). Their blocks of 2,280 bits were written with a checkerboard pattern of ‘0’s and ‘1’s in order to take the measurements of these blocks as the reference of zero aging as well as structured stored values. The other two ASICs were working continuously during 96 hours

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

TABLE X
ABSOLUTE PERCENTAGES OF CELLS CLASSIFIED AS STABLE (*A*) AND UNSTABLE (*B*) BEFORE AND AFTER AGING

	Normal	Accelerated aging with high Vdd				Accelerated aging with high T			
		Checker	Random	Best	Worst	Checker	Random	Best	Worst
A (before)	87.48% ±0.22%	87.15% ±0.24%	86.50% ±0.25%	86.45% ±0.29%	86.60% ±0.20%	87.08% ±0.15%	86.86% ±0.30%	86.56% ±0.24%	87.54% ±0.21%
A (after)	87.39% ±0.22%	87.69% ±0.22%	86.94% ±0.20%	86.86% ±0.25%	87.41% ±0.22%	87.35% ±0.16%	87.36% ±0.23%	87.05% ±0.24%	87.91% ±0.17%
B (before)	7.01% ±0.14%	6.79% ±0.19%	7.21% ±0.18%	7.30% ±0.20%	7.13% ±0.18%	6.96% ±0.17%	7.35% ±0.23%	7.43% ±0.18%	6.71% ±0.15%
B (after)	7.21% ±0.19%	6.71% ±0.18%	7.33% ±0.19%	8.21% ±0.20%	6.34% ±0.15%	6.75% ±0.22%	7.01% ±0.18%	8.29% ±0.22%	5.68% ±0.13%

under accelerating aging, one of them at increased power supply voltage (1.32V, 25ºC) and the other at increased temperature (1.2V, 75ºC), so that the influence of power supply voltage and temperature were analyzed separately like in the other experiments. To evaluate the influence of the values stored in the memory cells under accelerated aging, the 4 sets of 10 blocks were written as follows: (1) the first set was written with a checkerboard pattern of '0's and '1's; (2) the second with a random sequence of '0's and '1's; (3) in the third set, the *A* cells were written with the opposite of their start-up values and the *B* (and *C*) cells were written with their preferred values (the most repeated value in the 60 measurements done during classification); and (4) in the fourth set, the *A* cells were written with their start-up values and the *B* (and *C*) cells were written with the opposite of their preferred values. According to the initial classification, the values stored in the memory cells of the third set may reinforce the stability of *A* cells and the unstability of *B* cells. This is why that set is referred to as "best" in Tables X and XI. On the contrary, the fourth set is referred to as "worst" because the values stored may make *A* cells less stable and *B* cells less unstable.

The classification method was applied again after normal and accelerated aging operation. The initial and final percentages of stable and unstable cells were calculated. The mean values together with their standard errors are shown in Table X (per set of 10 blocks of 2,280 bits). The values before and after operation are similar. Only the percentage of *B* cells increase slightly in the "best" sets while decrease slightly in the "worst" sets. Hence, if registration of the SRAM is carried out at different time points during its lifetime and the long-term stored values are not carefully selected (as in the "best" or "worst" cases), similar percentages of stable and unstable cells are expected after classification.

The next analysis was to evaluate if the cells classified as *A* and *B* at the two registration phases were the same. Table XI shows the relative percentages of cells that were classified as *A* or *B* at the first registration and change or not at the last classification. The first column shows the results of the SRAM which worked in normal mode. A high percentage of *A* cells remain classified as *A* cells. The percentage of *B* cells that remain classified as *B* is smaller than in the case of *A* cells. As was shown in Figures 5a and 5b, with 20 measurements per Vdd, *B* cells are coarserly classified than *A* cells. What is important is that a non-significant percentage of *A* cells changes to *B* in the last classification, and, vice versa, the percentage of *B* cells that change to *A* is also negligible. An average of 0.10% of the cells in a block that change from *A* to *B* (a relative percentage of 0.11% of the cells initially classified as *A*) and 0.10% of the cells that change from *B* to *A* (a relative percentage of 1.49% of the cells initially classified as *B*) can be considered within the noise of the classification. This is important to ensure that, if a registered SRAM works in normal mode, *B* cells will not be used to generate IDs or secret keys and *A* cells will not be employed to generate random numbers. The cells that do not remain as *A* or *B* usually change to *C* cells.

Under accelerated aging, the percentage of *A* cells that remain *A* decreases and the percentage of *A* cells that change to *B* increases (more with high T than with high Vdd and more in the "worst" sets than in the "checker" or "random" sets). The decrease is higher in the percentage of *B* cells that remain *B* and the increase is higher in the percentage of *B* cells that change to *A*. The exceptions are the "best" sets. Under accelerated aging, the percentages of *A* that remain *A* and *B* that remain *B* are higher than in the other sets (they even increase with respect to the normal mode under accelerated aging with high Vdd). In addition, the percentages of *A* that become *B* and *B* that become *A* are smaller than in the other sets (they are even smaller than in the normal mode under accelerated aging with high Vdd).

Since the proposed method allows actualizing the registration of the SRAM embedded in its application context, possible misclassifications of cells due to aging can be avoided. In addition, the effect of aging can be reduced if *A* cells store the opposite of their start-up values and the *B* cells store their preferred start-up values.

V. CONCLUSIONS

The method presented simultaneously improves the generation of identifiers, secret keys and random numbers, which are security primitives required by many cryptographic applications. The generated identifiers are more repeatable, so their length can be shorter and more identifiers can be generated from the same memory. Also, the complexity of the error correcting codes is reduced, and the entropy of the random numbers generated is improved, so that the time and resources to generate secret keys and random numbers are

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

TABLE XI
CHANGES OF STABLE (*A*) AND UNSTABLE (*B*) CELLS UNDER AGING (IN RELATIVE PERCENTAGES)

	Normal	Accelerated aging with high Vdd				Accelerated aging with high T			
		Checker	Random	Best	Worst	Checker	Random	Best	Worst
A cells that remain *A*	98.32% ±0.09%	98.27% ±0.06%	98.12% ±0.09%	98.85% ±0.08%	97.74% ±0.01%	97.72% ±0.08%	97.76% ±0.08%	98.98% ±0.07%	96.39% ±0.17%
A cells that become *B*	0.11% ±0.02%	0.26% ±0.03%	0.17% ±0.02%	0.03% ±0.01%	0.31% ±0.03%	0.41% ±0.05%	0.48% ±0.05%	0.08% ±0.02%	0.99% ±0.10%
B cells that remain *B*	83.6% ±1.1%	78.60% ±0.80%	80.2% ±1.5%	85.02% ±0.56%	73.1% ±1.1%	72.8% 1.1%	71.6% ±1.3%	79.85% ±0.79%	61.7% ±1.1%
B cells that become *A*	1.49% ±0.27%	3.89% ±0.44%	3.08% ±0.46%	1.10% ±0.27%	5.22% ±0.58%	5.73% ±0.32%	7.83% ±0.77%	3.96% ±0.42%	12.65% ±0.79%

considerably smaller. The registration phase of the method can be repeated to improve the quality of the identifiers, secret keys, and random numbers generated at different operating conditions as well as to consider the memory aging.

The method is easy to implement with the memory embedded in its application context. It does not need to be performed by specialized vendors or in the factory where the memory is manufactured. A VLSI module with low area overhead can be included in the digital design to further facilitate the implementation of the method and increase the security of the realization since only non-sensitive information is stored and/or communicated outside the digital design.

ACKNOWLEDGMENT

The authors would like to thank M. C. Martínez-Rodríguez, E. Tena and M. A. Lagos for their help in the ASIC testing.

REFERENCES

[1] S. Kumar, J. Guajardo, R. Maes, G. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA," in *Proc. HOST*, Anaheim, CA, 2008, pp. 67–70.
[2] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic PUFs from flip-flops on reconfigurable devices," presented at the 3rd Benelux Workshop on Information and System Security (WISSec), Eindhoven, The Netherlands, Nov. 13-14, 2008.
[3] Y. Su, J. Holleman, and B. Otis, "A digital 1.6 pj/bit chip identification circuit using process variations," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 69–77, 2008.
[4] P. A. Layman, S. Chaudhry, N. J. G., and J. R. Thomson, "Electronic fingerprinting of semiconductor integrated circuits," US Patent 6,738,294, 2002.
[5] D. Holcomb, W. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. on Computers*, vol. 58, no. 9, pp. 1198–1210, 2009.
[6] G. Selimis, M. Konijnenburg, M. Ashouei, J. Huisken, H. de Groot, V. van der Leest, G. Schrijen, M. van Hulst, and P. Tuyls, "Evaluation of 90nm 6T-SRAM as physical unclonable function for secure key generation in wireless sensor nodes," in *Proc. ISCAS*, Rio De Janeiro, Brazil, 2011, pp. 567–570.
[7] G. Schrijen and V. van der Leest, "Comparative analysis of SRAM memories used as PUF primitives," in *Proc. DATE*, Dresden, Germany, 2012, pp. 1319–1324.
[8] S. Eiroa, J. Castro, M. Martinez-Rodriguez, E. Tena, P. Brox, and I. Baturone, "Reducing bit flipping problems in SRAM physical unclonable functions for chip identification," in *Proc. ICECS*, 2012, Seville, Spain, pp. 392–395.
[9] M. Claes, V. van der Leest, and A. Braeken, "Comparison of SRAM and FF PUF in 65nm technology," in *Information Security Technology for Applications*. Springer, 2012, pp. 47–64.
[10] A. Schaller and V. van der Leest, "Physically unclonable functions found in standard components of commercial devices," presented at 1st Workshop on Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE), Avignon, France, May 30-31, 2013.
[11] M. Bhargava, C. Cakir, and K. Mai, "Reliability enhancement of bistable PUFs in 65nm bulk CMOS," in *Proc. HOST*, San Francisco, CA, 2012, pp. 25–30.
[12] K. Xiao, T. Rahman, D. Forte, H. Y., M. Su, and M. Tehranipoor, "Bit selection algorithm suitable for high-volume production of SRAM PUF," in *Proc. HOST*, Arlington, VA, 2014, pp. 101–106.
[13] J. Guajardo, S. Kumar, G. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Proc. CHES*, Vienna, Austria, 2007, pp. 63–80.
[14] S. Okumura, S. Yoshimoto, H. Kawaguchi, and M. Yoshimoto, "A 128-bit chip identification generating scheme exploiting SRAM bitcells with failure rate of 4.45×10^{-19}," in *Proc. ESSCIRC*, Helsinki, Finland, 2011, pp. 527–530.
[15] M. Hofer and C. Boehm, "An alternative to error correction for SRAM-like PUFs," in *Proc. CHES*, Santa Barbara, CA, 2010, pp. 335–350.
[16] M. Cortez, A. Dargar, S. Hamdiou, and G.-J. Schrijen, "Modeling SRAM start-up behavior for physical unclonable functions," in *Proc. DFT*, Austin, TX, 2012, pp. 1–6.
[17] J. Kim, J. Lee, and J. Abraham, "Toward reliable SRAM-based device identification," in *Proc. ICCD*, Amsterdam, The Netherlands, 2010, pp. 313–320.
[18] F. H. Gebara, J. Kim, J. Schaub, and V. Strumpen, "Temperature-profiled device fingerprint generation and authentication from power-up states of static cells," US Patent 8,219,857, 2012.
[19] P. Koeberl, J. Li, R. Maes, A. Rajan, C. Vishik, and M. Wojcik, "Evaluation of a PUF device authentication scheme on a discrete 0.13um SRAM," *LNCS*, vol. 7222, pp. 271–288, 2012.
[20] M. Cortez, S. Hamdiou, 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, 2013, pp. 35–40.
[21] M. Bellido, A. Acosta, M. Valencia, A. Barriga, and J. Huertas, "Simple binary random number generator," in *Electronics Letters*, vol. 28, no. 7, pp. 617–618, 1992.
[22] D. Kinniment and E. Chester, "Design of an on-chip random number generator using metastability," in *Proc. ESSCIRC*, Florence, Italy, 2002, pp. 595–598.
[23] V. van der Leest, E. van der Sluis, G. Schrijen, P. Tuyls, and H. Handschuh, "Efficient implementation of true random number generator based on SRAM PUFs," in *Cryptography and Security: From Theory to Applications*. Springer, 2012, pp. 300–318.
[24] C. Tokunaga, D. Blaauw, and T. Mudge, "True random number generator with a metastability-based quality control," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 78–85, 2008.
[25] F. Veljkovic, V. Rozic, and I. Verbauwhede, "Low-cost implementations of on-the-fly tests for random number generators," in *Proc. DATE*, Dresden, Germany, 2012, pp. 959–964.
[26] C. Bösch, J. Guajardo, A. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," in *Proc. CHES*, Washington, DC, 2008, pp. 181–197.
[27] E. Barker and J. Kelsey, "NIST Special Publication 800-90A: Recommendation for random number generation using deterministic random bit generators," Computer Security Division, Information Technology Laboratory, Jan. 2012.
[28] J. Daugman, "The importance of being random: statistical principles of iris recognition," *Pattern Recognition*, vol. 36, pp. 279–291, 2003.

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2015.2471279

[29] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication, vol. 800-22, Revision 1a, April 2010.
[30] P. Brox, M. C. Martínez-Rodríguez, E. Tena, I. Baturone, and A. J. Acosta, "ASIC solution for mimo piecewise affine functions," *Int. Journal of Circuit Theory and Applications*, to be published, DOI: 10.1002/cta.2058.
[31] P. Brox, J. Castro, M. C. Martínez-Rodríguez, E. Tena, C. Jiménez, I. Baturone, and A. J. Acosta, "A programmable and configurable ASIC to generate piecewise-affine functions defined over general partitions," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 60, no. 12, pp. 3182–3194, 2013.
[32] N. Saxena and J. Voris, "Data remanence effects on memory-based entropy collection for RFID systems," *Int. Journal of Information Security*, vol. 10, no. 4, pp. 213–222, 2011.
[33] E. Tena, J. Castro, and A. Acosta, "Automatic and systematic control of experimental data measurements on ASICs," presented at 19th TC4 Symposium on Measurements of Electrical Quantities (IMEKO), Barcelona, Spain, July 18-19, 2013.
[34] B. Sunar, W. Martin, and D. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Trans. on Computers*, vol. 56, no. 1, pp. 109–119, 2007.
[35] Atmel: Innovative techniques for extremely low power consumption with 8-bit microcontrollers. (2006). [Online]. Available: http://www.atmel.com/images/doc7903.pdf
[36] Failure mechanisms and models for semiconductor devices - jep122g. (2011). [Online]. Available: http://www.jedec.org/standards-documents/docs/jep-122e
[37] M. Denais, V. Huard, C. Parthasarathy, G. Ribes, F. Perrier, N. Revil, and A. Bravaix, "Interface trap generation and hole trapping under NBTI and PBTI in advanced CMOS technology with a 2-nm gate oxide," *IEEE Trans. on Device and Materials Reliability*, vol. 4, no. 4, pp. 715–722, 2004.
[38] S. Mahapatra and M. Alam, "A predictive reliability model for PMOS bias temperature degradation," in *Proc. IEDM*, San Francisco, CA, 2002, pp. 505–508.
[39] R. Maes and V. van der Leest, "Countering the effects of silicon aging on SRAM PUFs," in *Proc. HOST*, Arlington, VA, 2014, pp. 148–153.

Iluminada Baturone received the 5-year (Hons.) degree and the Ph.D. (Hons.) degree in Physics from the University of Seville, Seville, Spain, in 1991 and 1996, respectively.

Since 1990, she has been with the Microelectronics Institute of Seville (IMSE-CNM) CSIC/University of Seville. She is also with the Dept. of Electronics and Electromagnetism of the University of Seville, where she is an Associate Professor since 2001. She has co-authored the books "*Microelectronic Design of Fuzzy Logic-Based Systems*" (CRC Press, 2000) and "*Fuzzy Logic-Based Algorithms for Video De-Interlacing*" (Springer, 2010) and more than 150 scientific papers. She has participated in more than 30 Spanish and European research and industrial projects, leading 5 of them. She holds 3 patents and is one of the developers of the *Xfuzzy* environment. Her current research interests include hardware security, microelectronic design of crypto-biometric systems, and neuro-fuzzy systems.

Miguel A. Prada-Delgado received the 5-year degree in Telecommunication Engineering (specialized in Electronics) from the University of Seville, Seville, Spain in 2013. In 2014 he obtained the Master degree in Microelectronics with honors due to his job named "*Unclonable identifiers and true random numbers generation from static memory cells*". He is currently pursuing the Ph.D. degree in hardware security at the Microelectronics Institute of Seville (IMSE-CNM) CSIC/University of Seville, thanks to a grant from *V Plan Propio de Investigación* through the University of Seville, Seville, Spain.

Since 2013, he has been with the Microelectronics Institute of Seville (IMSE-CNM) CSIC/University of Seville. Since 2014, he has also been with the Dept. of Electronics and Electromagnetism of the University of Seville.

Susana Eiroa received the 5-year degree in Telecommunication Engineering from the University of Vigo, Spain in 2007. She obtained the Master degree in Microelectronics in 2010 and the Ph.D. in Microelectronics with International Mention in 2014, both from the University of Seville, Seville, Spain.

She developed her Master thesis at IMEC, Leuven, Belgium, in 2007. From 2009 to 2015, she was with the Microelectronics Institute of Seville (IMSE-CNM) CSIC/University of Seville as a Ph.D. student and post-doctoral researcher, thanks to a grant from the *Junta de Andalucía*. Currently, she works at ALTER Technology TÜV NORD, Seville, Spain. She has participated in several R&D projects, has co-authored several international publications and holds a patent. Her main research area is hardware security, mainly focused on Physical Unclonable Functions (PUFS) and hardware attacks.

Copyright (c) 2015 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.