Countering the Effects of Silicon Aging on SRAM PUFs

Roel Maes
Intrinsic-ID
Eindhoven, The Netherlands
roel.maes@intrinsic-id.com

Vincent van der Leest
Intrinsic-ID
Eindhoven, The Netherlands
vincent.van.der.leest@intrinsic-id.com

***Abstract*—Silicon aging, in particular NBTI, causes many PUFs to exhibit a natural tendency of growing less reliable over time. This is inconvenient or even unacceptable for in-the-field applications. In case of SRAM PUFs it is observed that the impact of NBTI aging depends on the data stored in the SRAM. In this work, we investigate the effects of data-dependent silicon aging on SRAM PUF reliability under a number of realistic scenarios. In an accelerated aging experiment on a 65nm CMOS SRAM PUF implementation it is observed that many scenarios cause a smaller reliability reduction than natural aging. Some scenarios even show *anti-aging* effects, i.e. they cause the SRAM PUF to grow more reliable over time. This is a significant improvement when using an SRAM PUF. Even more so because data-dependent (anti-)aging has a particularly low overhead, requiring neither any changes to the PUF circuit nor any pre-deployment effort.**

I. Introduction

A physically unclonable function or PUF implemented on a silicon integrated circuit (IC) can be used as a hardware root-of-trust for the digital system running on that IC, e.g. to generate and store the system's master encryption keys. Such PUF-based key generators are increasingly being deployed in digital security products [1], [2], [3] since they often outcompete traditional non-volatile memories (e.g. Flash, EEPROM, anti-fuses, etc.) as highly secure yet efficient key storage solutions.

For such applications, a high-quality PUF is needed which is both unpredictable as well as reliable, i.e. PUF responses are random per instantiation but repeatable with limited noise over time and under all circumstances. It is well known that a PUF's operating conditions such as environment temperature and supply voltage affect its reliability. PUF implementations are therefore tested under varying conditions to determine their *worst-case reliability*, typically occurring at high temperature and voltage [4]. An application like a PUF-based key generator needs to deal with (and hence be designed for) this worst-case reliability to ensure a failure-free operation in the field.

Another variable affecting the behavior and hence the reliability of a PUF implementation is the effective lifetime of the IC. Certain physical phenomena in a silicon IC cause a circuit's parameters to slowly change over time, mostly degrading its performance and even leading to failures. The accumulated effect of these phenomena is called *silicon aging*. Because PUF responses take their randomness from minute process variations in the circuit's parameters, it is evident that silicon aging affects and usually also degrades a PUF's reliability over time [5]. A particular consequence hereof is that the worst-case reliability of a typical PUF construction is not only to be found at high temperatures and voltages, but also at a point in the future at the end of the device's lifetime, after years of silicon aging. This makes it non-trivial to estimate the worst-case reliability. Moreover, the required effort to obtain such an estimate often results in silicon aging being (inappropriately) ignored as a reliability degradation factor in evaluations of PUF implementations.

Based on physical models we can to a certain extent predict or simulate the expected effect of silicon aging. Another option is to run a silicon device at elevated temperature and voltage which accelerates the aging phenomena, and allows us to measure the effect after a significantly shorter amount of time. Based on these methods, an estimated guess of the worst-case reliability at the end of the device's lifetime can be made.

As an alternative to designing a PUF-based application for predicted future worst-case reliability, a radically different approach can be taken. As we will demonstrate in this work, for certain PUF constructions (in particular SRAM PUFs) it is possible to slow down, halt, or even reverse the effects of silicon aging on the PUF response reliability. The latter has the tremendous advantage that the PUF's worst-case reliability is then to be found at the beginning of the device's lifetime and can be measured immediately after manufacturing. Over time the PUF's reliability will now stay constant or even improve which means that the reliability requirements for the PUF-based application can be significantly relaxed, resulting in a gain in efficiency (e.g. less complex error-correcting codes).

Related Work: An SRAM PUF, as proposed by Guajardo et al. [6], is a PUF construction based on the power-up state of an SRAM array. SRAM PUFs have been thoroughly studied and invariably show high-quality PUF behavior, see e.g. [4], [7]. However, it was also shown by Maes et al. [5] that without precautions, SRAM PUF reliability does suffer from silicon aging. Bhargava et al. [8], [9] were the first to study the potential beneficial effects of silicon aging phenomena on the reliability of PUF structures, including SRAM PUFs. In [8] they demonstrate that a post-manufacturing burn-in stress (high temperature and high voltage) applied for 120 hours reduces the initial amount of bit errors of an SRAM PUF by 40%. However, such a long pre-deployment burn-in time does represent a large overhead in the manufacturing flow of a typical silicon IC. Moreover, once in the field their SRAM PUF is again subject to regular silicon aging and its reliability will deteriorate over time.

Our Contributions: In this work, we present:

- *Anti-aging* techniques for SRAM PUFs which are purely based on data-dependent silicon aging effects in regular SRAM cells during the regular lifetime of the IC. As a consequence, these techniques are directly usable on any standard SRAM (no circuit changes) without any pre-deployment overhead (no burn-in time).
- An overview and experimental validation (in 65nm CMOS) of a number of (anti-)aging scenarios differing only in the circumstances for generating the *anti-aging data*, and an assessment of their effect on the SRAM PUF's reliability over time.
- The identification of *ideal* anti-aging scenarios which effectively improve the SRAM PUF's reliability over the lifetime of the IC, with a reduction of 0.35% in the average amount of bit errors for the most optimal case.

Paper Outline: In Sect. II we provide some background on silicon aging, in particular the aging effects which enable data-dependent *anti-aging* for SRAM PUF cells. Next, in Sect. III we introduce a number of plausible (anti-)aging scenarios for SRAM PUFs and experimentally validate them on a 65nm CMOS test ASIC in an accelerated aging experiment. The most important findings of this experiment are discussed in Sect. IV and finally we conclude in Sect. V.

II. Background

A. PUF Quality Measures

The quality of a PUF is quantified by a number of experimentally verifiable measures. A PUF's *reliability* is measured by its average *intra-distance*, i.e. the (Hamming) distance between multiple evaluations of the same response on the same PUF instance. This is a good estimate of the number of bit errors one can expect in a response evaluation and is preferably very small. PUF response *uniqueness* on the other hand is measured by the average *inter-distance*, i.e. the distance between responses evaluated on different PUF instances. For binary response values, the average inter-distance is preferably very close to 50% of the response length. A PUF's *unpredictability* is ideally measured by evaluating the response *entropy*, but due to limitations on the amount of available experimental data this is often difficult to do in a meaningful manner. However, a necessary condition for a high response entropy is a low response bias which can be efficiently and accurately measured by calculating the average *(Hamming) weight* of the response vectors. Ideally the average response Hamming weight is also close to 50% of the response length.

B. Aging Effects in CMOS Silicon

The nominal operation of a silicon IC has a number of unintended but unavoidable side-effects which result in permanent physical alterations to the circuit's physical structure. Hence, an operational IC slowly but gradually changes over time, i.e. it *ages*. Eventually the induced physical changes affect the circuit's operation, typically in a degrading manner, and ultimately even lead to circuit failures after a long period. One dominant aging effect in modern ICs is negative-bias temperature instability (NBTI) causing a gradual increase in the threshold voltage, which is most evident in switched-on PMOS transistors. Other independent physical aging phenomena in CMOS devices are hot-carrier injection (HCI), time-dependent dielectric breakdown (TDDB) and electromigration (EM). For an overview of silicon aging effects we refer to [10].

C. SRAM PUFs and Data-Dependent (Anti-)Aging

An SRAM PUF [6] evaluates the power-up pattern of a standard 6T SRAM array. As shown in Fig. 1, at its core each SRAM cell in the array comprises two nominally matched CMOS inverters which are cross-coupled. Uncontrollable CMOS process variations introduce random parameter deviations which cause a mismatch between the inverter pairs affecting their power-up state. The predominant mismatch in an SRAM cell determining its power-up state is the difference between the threshold voltages (V_{th}) of both PMOS transistors P1 and P2. E.g. consider the case when random variations cause $V_{\mathrm{th,P1}}$ to be slightly smaller than $V_{\mathrm{th,P2}}$. As a result, at power-up (rising V_{dd}) P1 will start conducting before P2, causing A to go logically high and preventing P2 from switching on. The power-up state of the cell is hence $A = 1$. The larger the mismatch between $V_{\mathrm{th,P1}}$ and $V_{\mathrm{th,P2}}$, the stronger the power-up preference of a cell and hence the smaller the probability to power-up in its *wrong* state causing a PUF response bit error. Extensive experiments [4], [7] have demonstrated that due the independent random nature of process variations on each SRAM cell, the power-up pattern of an SRAM array demonstrates excellent PUF behavior, i.e. small intra-distances and both inter-distances and Hamming weights close to 50%.

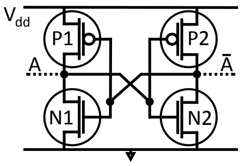


Fig. 1: Cross-coupled CMOS inverter circuit at the core of each SRAM cell. (Two SRAM access MOSFETs not shown.)

The effect of NBTI aging for SRAM cells depends on the bit value stored in the cell. When the cell stores a zero ($A = 0$), P1 is switched off and P2 is switched on. As a result, $V_{\mathrm{th,P2}}$ will increase over time due to NBTI while $V_{\mathrm{th,P1}}$ is unaffected. For $A = 1$ the opposite effect occurs. Combined with the power-up behavior, the situation is such that the PMOS with the smallest V_{th} tends to turn on at power-up and will subsequently experience a gradually increasing V_{th} due to NBTI. The natural tendency of an SRAM cell is hence to *age* such that $|V_{\mathrm{th,P1}} - V_{\mathrm{th,P2}}|$ grows smaller over time. From an SRAM PUF perspective, this is a disadvantage since a decreasing $|V_{\mathrm{th,P1}} - V_{\mathrm{th,P2}}|$ means a higher probability of a PUF response bit error. In other words, SRAM PUFs tend to become less reliable over time due to silicon aging, as was experimentally observed, e.g. in [5]. Luckily, this disadvantageous tendency can be counteracted. An evident solution is to let each cell store the inverse of its power-up value, since this would in general increase $|V_{\mathrm{th,P1}} - V_{\mathrm{th,P2}}|$ and hence make the corresponding SRAM PUF response bit effectively more reliable over time. This effect is called *anti-aging* in the context of (SRAM) PUFs.

III. EXPERIMENT: (ANTI-)AGING IN SRAM PUFS

A. Motivation

From Sect. II it is clear that the effect of silicon aging (in particular NBTI) on SRAM PUF reliability depends heavily on the (long-term) data stored in the SRAM, ultimately even permitting anti-aging. This observation gives rise to a number of interesting issues and questions worth investigating, e.g.:

- SRAM power-up states partially change between power-ups (intra-distance). Which particular state needs to be inverted to serve as anti-aging data? What happens when this exact state can not or only partially be reconstructed?
- When used as PUF response, SRAM power-up data is typically security-sensitive information. Which anti-aging options are still possible when all SRAM power-up information (including its inverse) needs to be erased?
- What are the (anti-)aging effects when the SRAM is used for other means after having served its PUF purpose?
- What is the effect of data-dependent (anti-)aging on other quality measures of SRAM PUFs, besides reliability?

These questions each represent different possible (anti-)aging scenarios for SRAM PUFs, which will be experimentally studied in this section.

B. (Anti-)Aging Scenarios

We start by listing the different (anti-)aging scenarios which we will test on a 65nm CMOS implementation of an SRAM PUF. For each scenario we detail the method used to generate the anti-aging data (i.e. the data which is long-term stored in the SRAM) and motivate the scenario by explaining how it can arise in a realistic use case. We consider the typical usage of a PUF in which an initial response measurement, called *enrollment*, is compared to or reconstructed from a later in-the-field measurement, called *reconstruction*. In the case of a key generator, the reconstruction measurement needs to be error-corrected to obtain the original enrollment response from which the key is derived.

1) No Anti-Aging: **NO_AA**

Scenario: The long-term data stored in the SRAM are the (noisy) SRAM power-up states at reconstruction.

Motivation: This is the trivial scenario where no action is taken (no anti-aging, no clearing, nor any other use of the SRAM), and the reconstruction power-up data is maintained unmodified in the SRAM array for the whole time the SRAM is powered. This scenario is mostly considered as a reference for the other scenarios. As derived in Sect. II, this is assumed to be the pessimistic scenario which fully exhibits the natural tendency of an SRAM PUF to grow less reliable over time.

2) Full Anti-Aging: **FULL_AA**

Scenario: Using error-correction techniques, the initial enrollment power-up state is perfectly reconstructed from a (noisy) reconstruction measurement. The long-term anti-aging data is the inverse of the corrected enrollment state.

Motivation: As derived in Sect. II, this is assumed to be the optimal scenario where the enrollment power-up state is continuously reinforced and the reliability of the SRAM PUF (w.r.t. the enrollment response) improves over time. This scenario can be applied straightforwardly in a PUF-based key generator which perfectly regenerates the enrollment response.

3) Partial Correction Anti-Aging: **PART_AA**$(xx\%)$

Scenario: Using error-correction techniques, the initial enrollment power-up state is *partially* reconstructed from a (noisy) reconstruction measurement, i.e. a certain fraction ($xx\% = 0\% \ldots 100\%$) of the bit errors in the reconstruction measurement is corrected. The long-term anti-aging data is the inverse of this partially corrected enrollment state.

Motivation: This is a variant of **FULL_AA** which describes the scenarios where for particular reasons the enrollment response is not perfectly reconstructed. For certain PUF-based applications it is often not required, inconvenient or impossible to perfectly reconstruct the SRAM enrollment power-up state, e.g. when the PUF response is used as a noisy identifier.

These first three scenarios (**NO_AA**, **FULL_AA**, **PART_AA**) are conceptually visualized in Fig. 2.

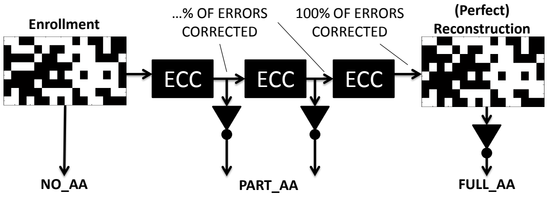


Fig. 2: Visualization of **NO_AA**, **FULL_AA**, and **PART_AA**.

4) Multiple Correction Anti-Aging: **MULT_AA**

Scenario: Using error-correction, one out of a set of initial enrollment SRAM power-up states is perfectly reconstructed from a reconstruction measurement. The long-term anti-aging data is the inverse of the selected enrollment state.

Motivation: This scenario arises when the same SRAM is enrolled as a PUF more than once, e.g. to derive a different PUF-based key for different users, applications, etc. Under normal circumstances, the different recorded enrollment states differ only due to noise on the PUF response bits.

5) Random Anti-Aging: **RAND_AA**

Scenario: The long-term anti-aging data is a random bit string. We differentiate between **RAND_AA(fix)** which uses a fixed random string after each reconstruction on each PUF instance, and **RAND_AA(true)** which uses a different randomly generated string after every reconstruction on every device.

Motivation: This is an (anti-)aging strategy that is independent of the power-up state. It avoids potential attack on security-sensitive data by erasing all PUF response information after it is used. Also it does not require error-correction.

6) *Structured Anti-Aging:* **STRUCT_AA**

Scenario: The long-term anti-aging data is a structured bit string, i.e. the anti-aging value of an SRAM cell is a deterministic function of the cell's memory address. The examples of specific data structures that we consider are:

1) **STRUCT_AA(zero)/STRUCT_AA(one)**: store zeros or ones in every bit location of the SRAM matrix.
2) **STRUCT_AA(row)**: alternately store zeros and ones in consecutive rows of the SRAM matrix.
3) **STRUCT_AA(checker)**: store a *checkerboard* pattern of zeros and ones in the SRAM matrix.

Motivation: Equal to **RAND_AA**, without the requirement for access to an embedded seeded PRNG or TRNG.

7) *Dynamic Anti-Aging:* **DYN_AA**

Scenario: After reconstruction, dynamically and constantly write data to the SRAM. We differentiate between **DYN_AA(rand)** which continuously writes randomly generated data, and **DYN_AA(fix)** which continuously cycles between a number of fixed patterns.

Motivation: This scenario arises when the SRAM used as a PUF, is also used for different purposes after key reconstruction, e.g. as an instruction or data memory. This scenario also captures a possible alternative (anti-)aging scenario which is independent of the power-up state.

C. Test Setup

As detailed in Sect. II, the major aging mechanism causing SRAM PUF reliability to change over time is NBTI. The effects of NBTI aging on a silicon device are considerably accelerated when the device is operated at increased temperature and/or supply voltage with respect to its nominal conditions [10]. The amount of acceleration is captured by an *acceleration factor* (AF) which depends on the accelerated aging conditions. For NBTI aging, AF is calculated as follows [10, Sect. 5.3]:

$$AF = \left(\frac{V_{stress}}{V_{nominal}}\right)^{\frac{\alpha}{n}} \cdot \exp\left(\frac{E_{aa}}{k} \cdot \left(\frac{1}{T_{stress}} - \frac{1}{T_{nominal}}\right) \cdot \frac{1}{n}\right) .$$

The used model parameters for NBTI accelerated aging are the following: the gate voltage exponent $\alpha = 3.5$; the time exponent $n = 0.25$; the apparent activation energy $E_{aa} = -0.02eV$; and Boltzmann's constant $k = 8.62 \times 10^{-5}eV/K$. The nominal and stressed aging conditions we consider in our accelerated aging experiment are as follows: $(T_{nominal}, V_{nominal}) = (40°C, 1.2V)$ and $(T_{stress}, V_{stress}) = (85°C, 1.44V)$. The resulting NBTI acceleration factor under these conditions becomes $AF = 18.6$, i.e. one hour of accelerated aging amounts to 18.6 hours of effective NBTI device aging under nominal conditions (assuming that AF is constant over time).

We applied this accelerated aging experiment on five 65nm CMOS test ICs each implementing four SRAM PUFs of size $8.0\mathrm{KByte}$. Prior to starting the aging experiment, several measurements on each of these 20 PUFs have been performed at $25°C$ ambient temperature and $1.2V$ supply voltage. These measurements represent the PUFs' initial states after zero weeks of aging. After these measurements, the temperature and voltage were increased to their stress levels. Once every week during the experiment the PUFs are remeasured at $25°C$ and $1.2V$ respectively to acquire a data set at nominal conditions for every week of accelerated aging. This accelerated aging experiment has run for 2856 consecutive hours.With an (assumed constant) acceleration factor of 18.6 this amounts to an effective NBTI aging of more than 6 years.

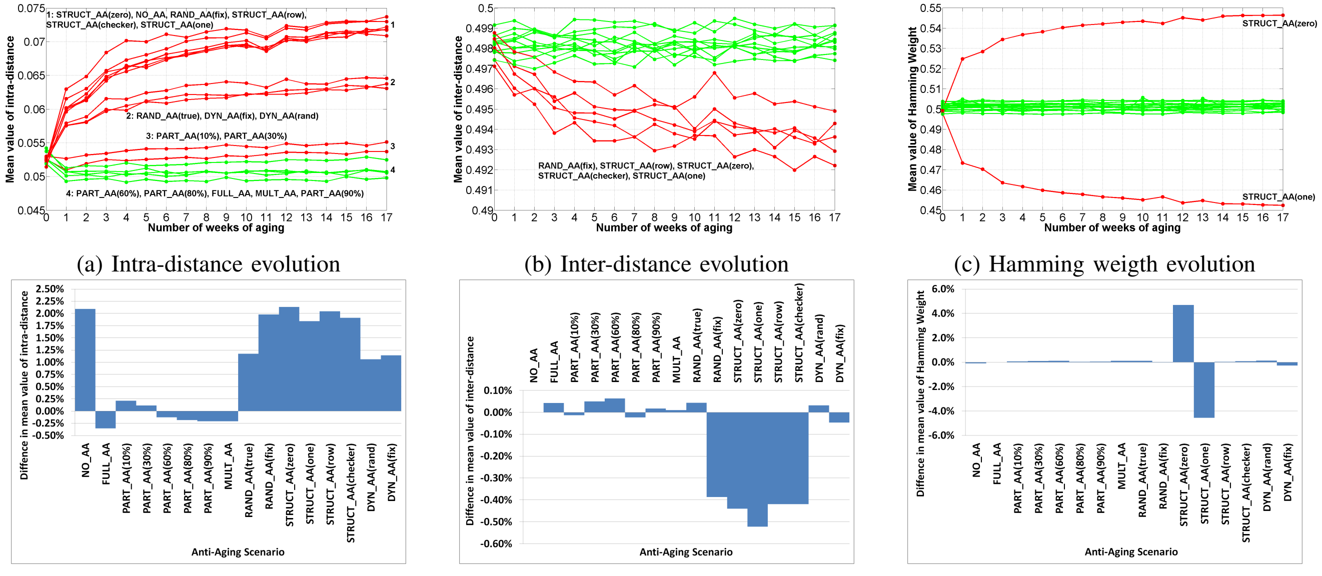
To evaluate the effect of the different (anti)-aging scenarios described in Sect. III-B, every SRAM PUF in the experiment is divided in 16 equal sections of size $0.5\mathrm{KByte}$ each implementing one of the scenarios. Every SRAM PUF is repowered, and hence re-evaluated, every six hours during the experiment to emulate a realistic usage, followed by the data actions prescribed by the different scenarios for each SRAM section. For most scenarios this means that the SRAM section is written with the scenario's data which remains in the memory for the next six hours of aging. For the **NO_AA** section no action is taken and the power-up data after every re-power is untouched. Special scenarios are **MULT_AA** for which a new inverse enrollment pattern is written every 80 minutes, and the dynamic scenarios **DYN_AA(rand)** and **DYN_AA(fix)** which are continuously overwritten. In Table I (which also already summarizes the results of these tests) an overview of the 16 sections and their respective (anti-)aging scenarios is presented. By evaluating the aging behavior of each of these sections individually, it is possible to analyze which scenarios are suitable for dealing with specific circumstances.

TABLE I: Considered (anti-)aging scenarios in our accelerated aging experiment, and the corresponding summarized results.
+ : quality measure is stable or improves (++ is best scenario)
– : quality measure degrades (–– is worst scenario).

#	Scenario	Written every...	Intra-distance	Inter-distance	Hamming weight
1	**NO_AA**	/	−−	+	+
2	**FULL_AA**	6 h	++	+	+
3	**PART_AA(10%)**	6 h	−	+	+
4	**PART_AA(30%)**	6 h	−	+	+
5	**PART_AA(60%)**	6 h	+	+	+
6	**PART_AA(80%)**	6 h	+	+	+
7	**PART_AA(90%)**	6 h	+	+	+
8	**MULT_AA**	80 min	+	+	+
9	**RAND_AA(true)**	6 h	−	+	+
10	**RAND_AA(fix)**	6 h	−	−	+
11	**STRUCT_AA(zero)**	6 h	−	−	−−
12	**STRUCT_AA(one)**	6 h	−	−−	−−
13	**STRUCT_AA(row)**	6 h	−	−	+
14	**STRUCT_AA(checker)**	6 h	−	−	+
15	**DYN_AA(rand)**	cont.	−	+	+
16	**DYN_AA(fix)**	cont.	−	+	+

D. Test Results

During the accelerated aging period the distributions of the PUF quality measures from Sect. II-A have been monitored: intra-distance (to evaluate reliability), inter-distance (uniqueness), and Hamming weight (unpredictability). The main focus is on intra-distance and the ultimate goal is finding *anti-aging*



(a) Intra-distance evolution

(b) Inter-distance evolution

(c) Hamming weigth evolution

(d) Intra-distance total change

(e) Inter-distance total change

(f) Hamming weight total change

Fig. 3: Impact on SRAM PUF quality measures during the accelerated aging test. The top graphs show the evolution of these measures during aging, while the bottom bar plots show their total relative change since the start of the experiment.

strategies which stabilize or even improve an SRAM PUF's reliability over time. However, the other two quality measures are also monitored to assure that any gain in reliability does not come at the expense of the PUF's uniqueness or unpredictability. A degradation of these parameters could cause a security risk since the PUF responses become less random.

An ideal anti-aging strategy prevents the mean intra-distance from increasing, but should also prevent the inter-distance and Hamming weight from moving away from 50%. For the tested SRAM PUFs, both the initial average inter-distance and Hamming weight are already very close to 50%, so they should preferably stay constant over time since any significant change (increase or decrease) makes them less ideal.

Using the output data of the experiment, a distribution of results from the 20 SRAMs has been compiled for each of the three PUF quality measures and for every week of aging elapsed in the experiment. The mean values (μ) of these distributions are calculated for each of the (anti-)aging scenarios. The evolution of these μ's for the different quality measures and aging scenarios for the duration of the experiment is presented in Figures 3a-3c [1], with colors indicating whether a quality measure is deteriorating over time (red) or not (green). The overall shift of the mean values for each of the quality measures during the total duration of the experiment is shown in Figures 3d-3f. A qualitative summary is to be found in Table I. The most important findings from these results and their consequences will be discussed in the next section.

IV. Discussion

The experimental aging results presented in Table I and Figure 3 clearly show that for reliability, the **NO_AA** scenario exhibits the worst-case evolution displaying the largest increase in intra-distance over time. **FULL_AA** on the other hand shows the best-case evolution with an intra-distance which even decreases over time. These experimental observations are completely in line with the hypothesized physical effects of NBTI aging for SRAM cells as detailed in Sect. II-C. We consider these two cases as references for assessing the performance of the other tested (anti-)aging scenarios.

1) Ideal scenario: Besides improving reliability over time by up to 0.35%, **FULL_AA** also has no negative effect on inter-distance and Hamming weight since it keeps them stable[2] at their already ideal values very close to 50%. In this respect, **FULL_AA** is to be considered as *the ideal anti-aging strategy* for SRAM PUFs and it is highly recommended to apply it whenever possible since it significantly relaxes the error-correction requirements of the PUF-based system. However, depending on the application and implementation constraints, the **FULL_AA** scenario is not always possible, e.g. because:

- a perfect error-free reconstruction of the enrollment data is not available;
- it is required that all (security-sensitive) PUF response data is completely erased after use, this rules out using its inverse as anti-aging data;
- the SRAM is not solely dedicated to the SRAM PUF, but is used for other purposes afterward;

[1] These graphs show a slowing down of most (anti-)aging effects on the SRAM PUF toward the end of the accelerated aging experiment which could be an indication that the accelaration factor AF as determined in Sect. III-C is not constant but slowly decreasing over time.

[2] We consider very small changes of less than 0.10% as being stable since they are not statistically significant.

- the SRAM PUF is enrolled multiple times and there is no single ideal enrollment data pattern.

These constraints have led us to investigate the (anti-)aging effects of other data scenarios, which we discuss next.

2) No/partial error-correction: When a perfect reconstruction of the enrollment data is not possible, the preferable scenario becomes **PART_AA**. Depending on the amount of achievable error-correction, the effect on reliability varies (the more error-correction, the better the effect on reliability). The results show that with a limited error-correction of 30% or lower, the reliability will slowly degrade over time. However, the deterioration is very small and significantly less than for **NO_AA**. With more error-correction the reliability will improve over time, as the results for 60% correction and higher show. Similar to **FULL_AA** the inter-distance and Hamming weight remain stable.

3) PUF zeroization: In case all security-sensitive PUF response information needs to be deleted from the SRAM, the best performing scenario is **RAND_AA(true)**. Although reliability degrades over time, it is still the most reliable scenario in which the data stored in SRAM is not based on the PUF response. The increase in intra-distance is also still considerably smaller than for **NO_AA**. The inter-distance and Hamming weight remain stable in this scenario. An important remark here is that the "obvious" option of really zeroizing the SRAM, i.e. overwriting it with all zeros (or ones), is a particularly bad choice since it not only degrades the reliability worse than all other options (besides **NO_AA**), it also has a significant negative impact on both inter-distance and Hamming weight. In fact, due to their negative effect on both intra- and inter-distance, none of the structured scenarios (**STRUCT_AA(...)** as well as **RAND_AA(fix)**) are recommendable, and one should even take care to avoid them in in-the-field situations.

4) Non-dedicated SRAM: When the SRAM is used for other purposes besides the SRAM PUF, the **DYN_AA** scenarios come into play. In these scenarios the reliability decreases over time, which is as expected because the data in the SRAM is not related to the PUF data. However, the reliability is still significantly better than for **NO_AA**, and when the dynamically written data to the SRAM is sufficiently random it will not deteriorate the inter-distance and Hamming weight.

5) Multiple-enrolled PUF: The **MULT_AA** scenario applies to SRAM PUFs which are enrolled more than once. The performance on all three quality measures is very good and comparable to that of **FULL_AA** and **PART_AA** with high error-correcting capabilities. **MULT_AA** exhibits an improvement in reliability and hence anti-ages the SRAM PUF.

V. Conclusion

We investigated the effects of silicon aging (NBTI) on SRAM PUF reliability. The physics behind NBTI predicts that SRAM PUFs have a natural tendency to become less reliable over time, but also reveals a potential solution which could even *anti-age* SRAM PUFs using the appropriate data-dependent actions. These cases, as well as a number of other relevant scenarios were tested in an accelerated aging experiment on a set of SRAM PUFs implemented on 65nm CMOS ICs. The test results confirm the predictions and identify an ideal anti-aging strategy which causes the tested PUFs to become 0.35% more reliable over an aging period of more than 6 years, without degrading the other PUF quality measures. This *full anti-aging* strategy even improves SRAM PUF reliability up to 2.45% compared to the natural reliability degradation over the same period when no actions are taken. Even when the full anti-aging scenario is not applicable, there are still strategies which are significantly better than doing nothing. A major practical advantage of the proposed anti-aging solutions is that they do not require any circuit changes or pre-deployment effort. They are hence usable for standard SRAM implementations in a regular development flow.

A noteworthy side observation of independent interest is that SRAM PUF zeroization (for security reasons) with a fixed pattern (e.g. all zeros) is particularly detrimental for each of the PUF's quality measures, and therefore highly discouraged. Overwriting the PUF response with on-the-fly randomly generated data is preferable instead.

Acknowledgment

We would like to thank Peter Simons and Dariusz Tesmer for their valuable help in performing the tests. This work has been supported in part by the European Commission through the FP7 programme under contract 284833 PUFFIN, and through the Catrene project RELY. The ASIC used in the described tests was developed in the FP7 project UNIQUE.

References

[1] NXP, "SmartMX2 unleashes secure multi-applications without compromise," http://www.nxp.com/documents/leaflet/75017276.pdf, Aug. 2012.

[2] Microsemi, "SmartFusion2 SoC FPGA Reliability and Security User Guide," http://www.microsemi.com/document-portal/doc_download/130926-smartfusion2-security-and-reliability-user-s-guide, Sep. 2013.

[3] Coherent Logix, "HyperX: Security Processing and Trusted Computing," http://www.coherentlogix.com/products/hyperx-processors/security/, Jan. 2014.

[4] S. Katzenbeisser, U. Kocabaş, V. Rožić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon," in *Cryptographic Hardware and Embedded Systems (CHES) 2012*, ser. LNCS. Springer, 2012, vol. 7428, pp. 283–301.

[5] R. Maes, V. Rožić, I. Verbauwhede, P. Koeberl, E. van der Sluis, and V. van der Leest, "Experimental evaluation of Physically Unclonable Functions in 65 nm CMOS," in *European Solid-State Circuits Conference (ESSCIRC)*, Sep. 2012, pp. 486–489.

[6] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Cryptographic Hardware and Embedded Systems (CHES)*, ser. LNCS, vol. 4727. Springer, 2007, pp. 63–80.

[7] G.-J. Schrijen and V. van der Leest, "Comparative analysis of SRAM memories used as PUF primitives," in *Design, Automation Test in Europe (DATE) 2012*, 2012, pp. 1319–1324.

[8] M. Bhargava, C. Cakir, and K. Mai, "Reliability enhancement of bi-stable PUFs in 65nm bulk CMOS," in *Hardware-Oriented Security and Trust (HOST)*, 2012, pp. 25–30.

[9] M. Bhargava and K. Mai, "A High Reliability PUF Using Hot Carrier Injection Based Response Reinforcement," in *Cryptographic Hardware and Embedded Systems (CHES)*, ser. LNCS. Springer, 2013, vol. 8086, pp. 90–106.

[10] JEDEC, "Failure Mechanisms and Models for Semiconductor Devices - JEP122G," http://www.jedec.org/standards-documents/docs/jep-122e, Oct. 2011.