Special Session: Security Verification & Testing for SR-Latch TRNGs

Javad Bahrami*, Mohammad Ebrahimabadi*, Jean-Luc Danger†, Sylvain Guilley†‡, Naghmeh Karimi*

*University of Maryland Baltimore County, United States
†LTCI, Télécom Paris, Institut Polytechnique de Paris, France
‡Secure-IC S.A.S., France

***Abstract*—Secure chips implement cryptographic algorithms and protocols to ensure self-protection (e.g., firmware authenticity) as well as user data protection (e.g., encrypted data storage). In turn, cryptography needs to defer to incorruptible sources of entropy to implement their functions according to their mandatory usage guidance. Typically, keys, nonces, initialization vectors, tweaks, etc. shall not be guessed by attackers. In practice, True Random Number Generators (TRNGs) are in charge of producing such sensitive elements.**

Fully aware of the central role of TRNGs in the proper implementation of security in chips, stakeholders have been formalizing the requirements recently. The methods to strengthen such requirements are manifold. In this paper, we discuss and apply three of them by targeting the Set-Reset Latch TRNG which is an alternative to Ring-Oscillator (RO) TRNGs as it provides faster throughputs. The first method concerns the confidence in the TRNG being random enough. It explores how the TRNG properties can be reliably predicted by simulation, compared to real silicon experiments. The second aspect dealt with in this paper is the assessment of the TRNG properties over time, i.e., considering the impact of aging in the TRNG properties. Such knowledge is important as secure chips are expected to be in service for a long period, and it would be detrimental to the service they render if the quality of the entropy they deliver would be declining over time. Eventually, the third aspect of this paper is the timely detection of unforeseen failures or malevolent attacks. The mitigation lies in leveraging "health tests" launched prior to using random numbers.

This paper focuses on a particular type of TRNG that is not prone to biasing by attackers: it is the so-called Set-Reset Latch (SR-latch) TRNG and exploits a race condition in an arbitration gate. Such kind of TRNG is of great practical interest as an alternative design compared to the mainstream "Ring Oscillator" TRNG, and it is also very amenable to analyses by various sorts of simulations aiming at properly characterizing its security in various operational environments.

***Index Terms*—True Random Number Generators (TRNGs), Set-Reset Latch TRNG, Security, Verification & Testing, Aging, Health tests, Compliance to standards.**

I. Introduction

Secure chips require a source of uniformly distributed bit-strings. Such source allows to generate all the random parameters required to implement cryptographic protocols properly. Indeed, most, if not all, cryptographic algorithms implicitly rely on the availability of fresh random values. For instance, AES-CBC initialization vectors, HMAC keys, ECDSA nonces, Crystals Kyber noise, etc. all need to be provided by "True Random Number Generators" (TRNGs). Those sources of entropy should be independent such that an attacker cannot get to know or influence them. For this reason, hardware TRNGs have been put forward.

Ring-Oscillator based TRNGs (RO-TRNG) have been out for a while, and are trustworthy. However, it is not enough to only rely on RO-TRNGs, as based on current security assurance requirements there is a need for an alternate TRNG design as well to avoid *single point of failures* (SPOFs). There are various TRNGs based on other rationales. However, there is an obviously advantage for designs which are integrable in CMOS logic, and another bonus for PPA-efficient[1] structures. Accordingly, in this paper, we tackle one such TRNG, i.e., Set-Reset latch (SR-latch) based TRNG. The main concern here is thus to analyze the randomness of such security primitive not only when these primitives are new but also when they have been used for a while.

II. In silicon validation of SR-latch TRNG

A. Principle of SR-latch TRNG

A TRNG designed in digital devices exploits either the noise coming from clock jitters, the noise around metastable states, or both. The exploitation of the phase jitter noise is carried out by means of oscillators, like ROs, and represents a robust way to generate randomness. The use of metastable state is more tricky as it requires analog and custom cell [10] but it is much faster than the oscillator-based TRNG. A proposal to build such TRNG in full digital environment is to use latches like in [6], [11] which exploit both the jitter and metastable state. The use of NOR-based SR-latch, represented in Figure 1, allows to capture the noise around a metastable state.

The Set (S) and Reset (R) inputs are derived by the same signal SR. If the two NOR gates are perfectly balanced, when the SR input goes to zero, the latch would be in a metastable state at around $V_{dd}/2$ voltage. However, the environmental noise at the input creates a convergence towards a stable state (either V_{dd}, interpreted as logic '1', or 0 V, interpreted as logic '0'). This is depicted in the simulations shown in Figure 2, showing the values of Q (output of the SR-latch) over 100

[1]PPA stands for "Performance, Power and Area", three important figures of merit (aside Safety & Security) in hardware designs.

979-8-3503-4630-5/23/$31.00 ©2023 IEEE

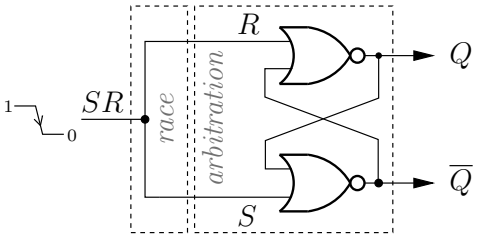

Fig. 1. An SR-latch realized via cross-coupled NOR gates.

repetitions. In this figure, the value of $V_{dd} = 1.2$ V. It can be seen that it takes a certain time for the final value to converge and its entropy depends on the amount of the physical noise.

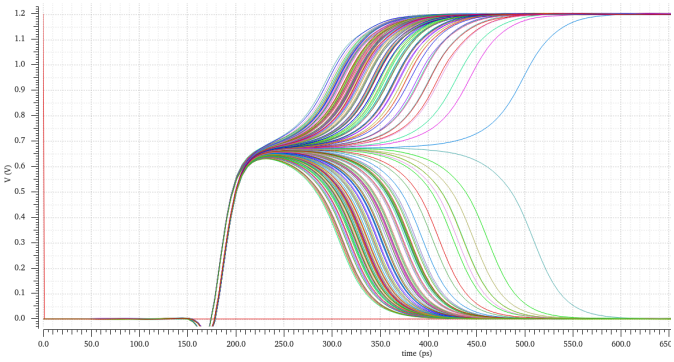

Fig. 2. Transition from a metastable to a stable state due to input noise.

The metastability, shown to several hundreds of picoseconds, can be resolved by adding a synchronizing flip-flop, clocked by SR) at the output Q or $\bar{Q}$ of the SR-latch. Henceforth, the SR-latch TRNG can be operated with a periodic signal SR which can be the system clock. Thus one random bit is gotten at every clock period, which makes the entropy source at the same time fast and compact (hence economic in terms of PPA; referring to Power-Performance-Area).

As discussed, a latch can be used as a **TRNG**, as its outcome depends on the dynamic noise. Note that if the two NOR gates are strongly unbalanced, due to *locally correlated* process mismatch, the latch will always converge to the same stable state. In this case the latch behaves like a "Physically Unclonable Function" (also known as **PUF** [12]), as its value depends on the fabrication noise which has become static. To enhance the probability of getting a true random output, it is necessary to consider a set of latches; instead of only one latch. Figure 3 illustrates the architecture where the embedded latches' outputs are XORed together to realize a TRNG.

B. Stochastic model of the SR-latch TRNG

A stochastic model can be built to assess the entropy of the SR-latch TRNG according to the number of latches and the process mismatch. Indeed, the XOR combination of latches can give an entropic random number if there are a minimum number of latches around a metastable state and if the process mismatch is not too high. A first approach has been

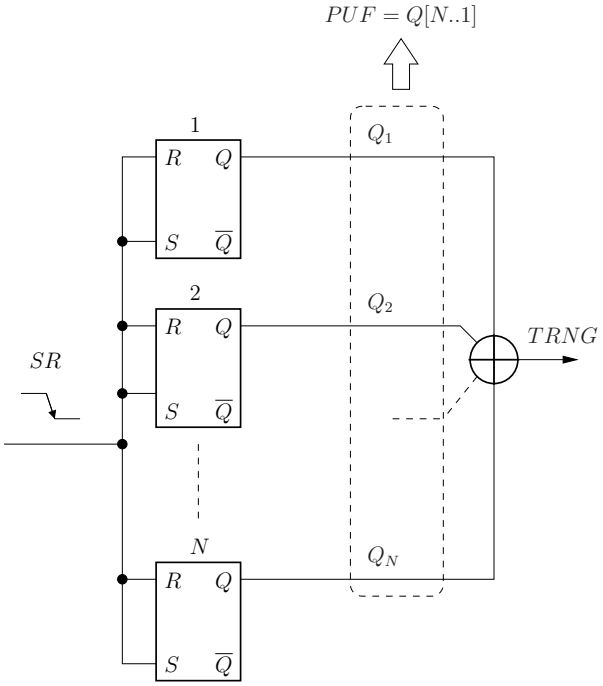

Fig. 3. The PUF-TRNG circuitry composed of a set of SR-latches.

studied in [4] but did not propose a closed form expression to assess the entropy and the number of latches according to the mismatch.

Every latch i outputs a random variable Q_i which can be considered as a Bernoulli random variable associated with a probability $\mathbb{P}[Q_i = 1] = p_i$. The XOR between these variables can be obtained by changing the variable $p_i \in [0, 1]$ by $q_i \in [-1, 1]$, such that:

$$p_i = \frac{1 + q_i}{2}, \qquad q_i = 2p_i - 1.$$

The probability of the XOR between two latches Q_i and Q_j is thus:

$$\mathbb{P}[Q_i \oplus Q_j = 0] = p_i \cdot p_j + (1 - p_i)(1 - p_j) = \frac{1 + q_i q_j}{2}.$$

By induction (refer to the *piling-up lemma* [14]), one gets that the resulting probability of the node $TRNG$ with the XOR of N latches is equal to:

$$P_0 = \mathbb{P}[TRNG = 0] = \frac{1 + \prod_{i=1}^{N}(2p_i - 1)}{2}\ ,\ P_1 = 1 - P_0. \tag{1}$$

The probability p_i highly depends on the imbalance between the S and R input. This bias comes from mismatches in the process variation and the routing. It is equivalent to a delay offset Δ_M which has a Gaussian distribution: $\Delta_M \sim \mathcal{N}(0, \Sigma^2)$. With respect to the noise, Z being also considered Gaussian $Z \sim \mathcal{N}(0, \sigma^2)$, we define the "Mismatch to Noise Ratio" MNR as being:

$$\text{MNR} = \frac{\Sigma}{\sigma}\ . \tag{2}$$

The mean probability of p_i according to MNR is given in Equation 3, and it can be deduced from the section III.A of the article from Schaub et al. [19]. This study considers the SR-latch as a PUF and thus gives the Bit Error Rate (BER) probability which is fully equivalent to the p_i probability when the SR-latch is used as a TRNG.

$$\widehat{\mathrm{p_i}} = \frac{1}{\pi} \arctan\left(\frac{1}{\mathrm{MNR}}\right). \tag{3}$$

If we use N SR-latches in parallel, Equation 1 applies. If we consider the noise independent from one latch to another, The mean probability is given by Equation 4:

$$\widehat{P_0} = \frac{1 + (2\widehat{p_i} - 1)^N}{2} = \frac{1 + (\frac{2}{\pi} \arctan(\mathrm{MNR}))^N}{2} \tag{4}$$

and, by complementarity,

$$\widehat{P_1} = 1 - \widehat{P_0}.$$

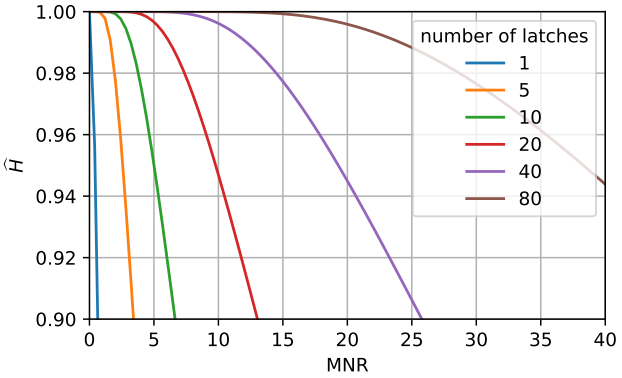

Fig. 4. Mean Entropy $\widehat{H}$ according to Mismatch to Noise Ratio (MNR) for different number of latches.

The mean Shannon entropy is given by Equation (5):

$$\widehat{H} = -\sum_{i \in \{0,1\}} \widehat{P}_i \log(\widehat{P}_i). \tag{5}$$

It is illustrated in Figure 4 according to the number of latches and the MNR parameter.

This figure shows that it is necessary to have a significant number of SR-latches to ensure a good entropy, especially if the technology has a high process variance. The study presented in the next section III gives some information about the MNR quantity in the current technology.

III. Tests on an ASIC version of the SR-latch TRNG

A. Architecture

A design with 1024 SR-latches has been implemented in a testchip fabricated in 28nm FD-SOI process and presented in [7]. As a primary goal, it has been built to demonstrate the impact of the body-bias voltage on the properties of a PUF-TRNG architecture. In order to drive the latches from outside

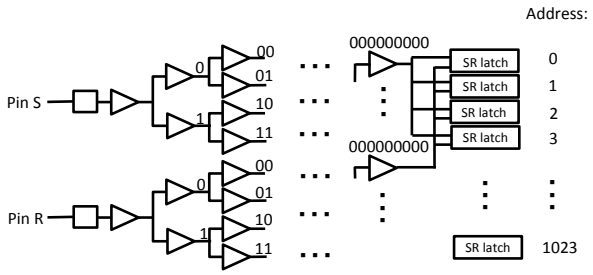

Fig. 5. Structure of the two buffer trees of the PUF-TRNG testchip.

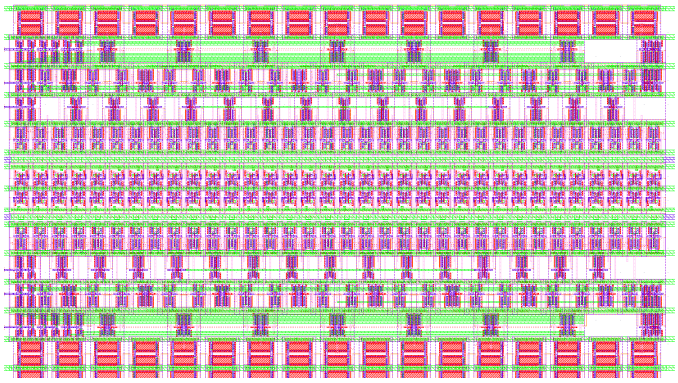

Fig. 6. Partial layout of the PUF-TRNG circuit.

of the chip, a buffer tree is added on each S and R input which have respective input pins, as represented in Figure 5.

The trees have been carefully balanced to guarantee the simultaneous arrival time of the S and R signals to all the latches. But as the buffer trees introduce a higher sensitivity to process mismatches between the S and R inputs, the architecture is in favor of PUFs (i.e. stable outputs of latches). On the opposite, the imbalance generated by the buffer trees could decrease the number of metastable latches necessary for the TRNG.

Figure 6 shows a part of the layout. The latches are placed in the middle rows whereas the buffers are placed in a balanced manner at the top and the bottom of the latches

The NOR gates are built using standard regular threshold voltage V_{th} transistors, but the PMOS transistors of the top NOR and bottom NOR have different and controllable reverse body bias, denoted $VB1$ and $VB2$ respectively. It is shown in [7] that the range of $VB1, VB2$ is quite large to get a high number of SR-latches either stable for PUF, or unstable for latches for TRNG. Another takeaway result is that the optimal pair $(VB1, VB2)$ where there is a maximum of latches stable for PUF happens to be exactly the same as the optimal pair $(VB1, VB2)$ where there is a maximum of latches unstable for TRNG.

B. Evaluation of the Mismatch to Noise Ratio

The delay Δt between the S and R input is swept to measure the number of stable and unstable latches, more precisely to get the probability p_i of every latch. It has to be noted that

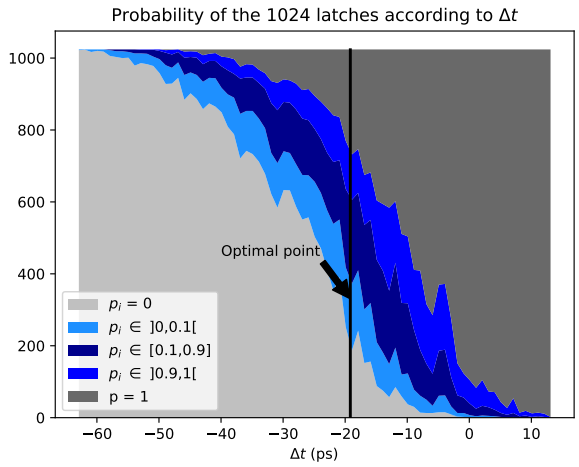

Fig. 7. Distribution of the 1024 latches output value probability (p_i) *vs.* Δt.

Δt is relatively important (a few ps) as there is an imbalance between the paths of the S and R signals. This imbalance results from the placement and routing of the pins and the stages of the buffer tree. In Figure 7 we can see the proportion of latches stable with p_i at 0 or 1, and unstable latches with $p_i > 0$ and $p_i < 1$.

At Δt around -20 ps, the mean probability $\widehat{P_1}$ is assessed at 0.045, leading to an MNR estimated at around 7 according to Equation 3. As there is a high mismatch coming from the buffer tree, the real MNR of SR-latches having S and R connected together as in Figure 1 is certainly less than 7.

With $MNR = 7$, if one wants to get an entropy equal to 0.997 bit (as mandated by AIS31 standard [17]), thirty (30) SR-latches would be necessary according to equation 5. Figure 7 also shows that if Δt is not exactly at -20ps, the MNR value stays almost the same. That means that there is a tolerance if the internal delay offset is biased, meaning that Δ_M introduced in section II-B may not be necessarily centered.

IV. Predicting the Effect of Aging on TRNG

As the transistors' specification change over time due to the so-called device aging, we need to investigate how the SR-latch TRNG behavior evolves over time. Accordingly, in this section, we first give a brief review of device aging, its causes and impacts, and then we present our experimental results showing how aging affects the TRNG outcome over time.

Indeed in this study, the open question is that due to aging which of the following cases may happen?

- *Less Randomness* because of independent aging of the two cross-coupled NOR (more unbalancedness); or
- *More/less aging* on the PMOS of the NOR gate that was faster when the device was new (not aged);

If the second case applies in a negative sense, is it possible to take advantage of the differential aging of the 2 NOR gates to make up for aging by recovering? Indeed, by setting some values in the SR-latch at rest, can a bias be compensated, as in the case of PUF (see [8], [15])?

A. Background on Device Aging

Aging mechanisms result in performance degradation and eventual failure of digital circuits over time. In CMOS technology, the two leading factors of aging are Bias Temperature Instability (BTI) and Hot Carrier Injection (HCI) [2]. Both mechanisms result in increasing switching voltage and path delays. NBTI (one class of BTI) [2] affects PMOS transistors, while PBTI (another class of BTI) as well as HCI affects NMOS devices.

BTI Aging: A PMOS (NMOS) transistor goes under two phases of NBTI (PBTI) depending on its operating condition [2]. The first phase, i.e., $stress$, occurs when the related transistor is "ON". Here, charges are trapped at the Si-SiO_2 interface and lead to an increase in the threshold voltage. The second phase, $recovery$, occurs when the transistor is off. In this phase, the charges trapped in the stress phase are partially removed, and thus the threshold voltage (V_{th}) drift that occurred during the stress phase partially recovers. The impact of BTI depends on the supply voltage, temperature, physical parameters of the transistor under stress, and stress time. Fig. 8 depicts the V_{th} drift of a PMOS transistor when it is continuously under stress for 6 months versus the case that it experiences stress and recovery phases every other month. In this figure, the values on the Y-axis are not shown intentionally to make the figure generic and technology independent.

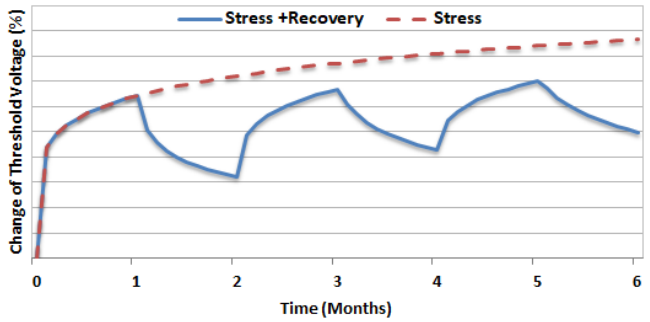

Fig. 8. NBTI-induced V_{th} drift of a PMOS transistor that is always under the stress, and a transistor that is under the stress and recovery alternatively.

HCI Aging: HCI happens in an NMOS when hot carriers are injected into the gate dielectric during transistor switching and remain there. HCI is a function of switching activity; degrading the circuit by shifting the threshold voltage and drain current of stressed transistors. The threshold voltage drift induced by HCI depends on the activity factor of the transistor under stress, its temperature, clock frequency, and usage duration [16].

B. Experimental Setup

To answer the above question and study the impact of aging on the SR-Latch-based TRNGs, we conducted extensive Spice

simulations. We implemented our TRNG at the transistor level using a 45nm open-source NANGATE library [1]. Our netlist includes 20 SR-latches in parallel (following Fig. 3). To mimic the real-silicon behavior, we considered process variation (PV) through Monte-Carlo simulations with Gaussian distributions: transistor gate length L: $3\sigma = 10\%$, threshold voltage V_{TH}: $3\sigma = 30\%$, and gate-oxide thickness t_{OX}: $3\sigma = 3\%$. This allows us to derive the delay offset of the mismatch Δ_M as defined in section II-B.

In our implementation of the SR-latch TRNG, while the $\overline{Q}$ outputs of all latches are left floating, the Q outputs are XORed together to build the final single bit of randomness. We benefited from Synopsys HSpice for the transistor-level simulations, and the HSpice built-in MOSRA Level 3 model [20] to extract NBTI, PBTI, and HCI aging effects. The effect of aging was evaluated for 7 years of device operation in time steps of 6 months. The simulations were conducted for temperature of 85°C, $V_{dd} = 1.2$ V, and the operating frequency of 500 MHz.

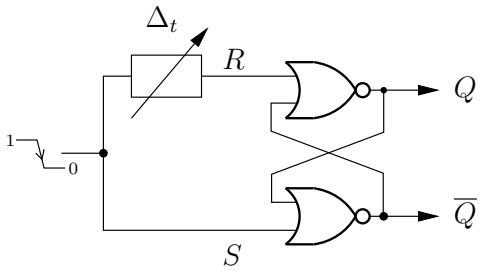

Fig. 9. SR-latch with a controlled Δ_t to study the mismatch and noise impact.

In a perfect netlist, the *R* and *S* signals arrive simultaneously, hence the latches get metastable. However, as there is a mismatch in processes and wiring, the outputs of latches might go to a known state and the entire circuit behaves as a PUF. For instance, PMOS transistors in one of the NOR gates residing in the latch might have a lower threshold voltage V_{th} and get evaluated faster than the other ones (PV). Bulk and flicker noise (specifically in low frequencies) are other important sources of unwanted behavior in CMOS technology that might cause the circuit to behave predictably.

As modeling all these effects separately is not trivial, we abstract them away and add a controlled delay Δt to the *R* signal of each embedded latch while *S* signals are assumed to be clean as shown in Figure 9.

C. Analyzing the Process Variation Induced Delay

To assess the impact of process variation in our latches, we extract the time Δ_M at which the output of each of the embedded latches (shown in Fig. 3) toggles. To do so, we sweep the delay Δt shown in Figure 9 in the *R* signal path towards the *S* signal with the step size of 80 fs. With such sweeping, the latch goes first to the metastable state (red area in Figure 10) and then the output is toggled with more sweeping. The exact moment of Δ_M is when our latch toggles with the accuracy of 80 fs similar to the sweeping step size.

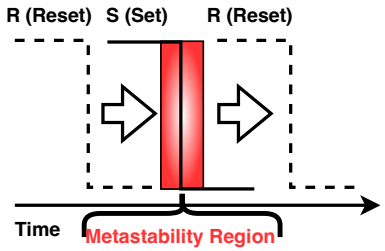

Fig. 10. Sweeping Δt of the *R* signal with step size of 80 fs.

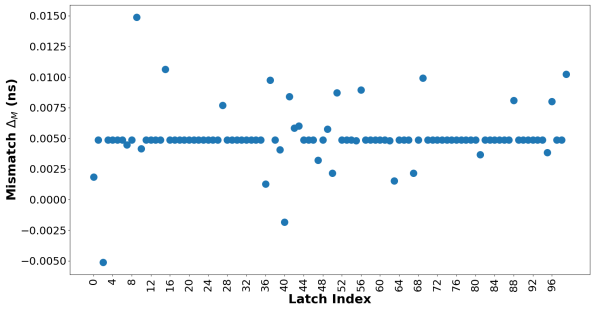

Fig. 11. Mismatch distribution (ns) of the circuit with 100 latches.

We collect this data for each of the embedded latches and use this data for the statistical analysis discussed below. Such information provides a precise insight into the statistical behavior of SR-latch-based TRNGs and helps us to answer the following questions:

- What is the value of Δ_M for every latch?
- What is the mean and variance of Δ_M distribution (denoted as μ and Σ^2, respectively)?

Figure 11 shows the results of the analysis for a TRNG realized via 100 parallel latches. The takeaway points from this set of results include:

- The output of the majority of the embedded latches flips while the *S* and *R* signals are 5 ps apart from each other ($\mu = 5$ ps) due to the architectural bias we have in our implementation. Hence, Δ_M is not centered, $\Delta_M \sim \mathcal{N}(\mu, \Sigma^2)$, and the stochastic model proposed in section II-B may be optimistic even if the ASIC results in section III show that this bias is not problematic in the circuit designed in FD-SOI.
- Figure 12 shows the distribution of Δ_M which should be near a Gaussian distribution. The estimated standard deviation is $\Sigma = 2$ ps.

We repeated the experiments with 20 latches, this time by adding a random Gaussian delay on Δ_t of the *R* signals with $\sigma = 0.25$ ps and $\sigma = 3$ ps. With the knowledge of the delay offset Δ_M coming from the mismatch, the noise standard deviation $\sigma \in \{0.25, 3\}$ ps corresponds to the mismatch to noise ratio MNR of $\{8, 2/3\}$, respectively. Thus in case of considering the XOR of 20 latches, we can deduce from

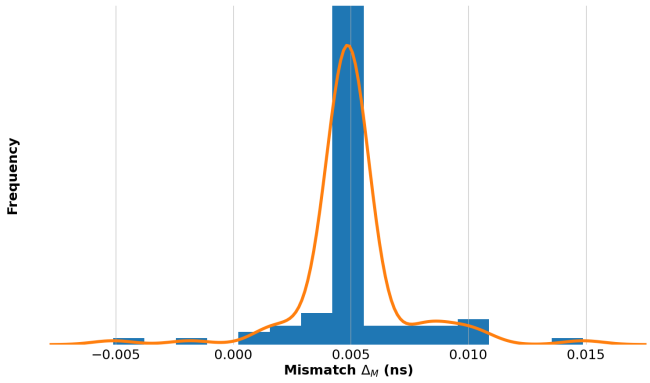

Fig. 12. Mismatch distribution histogram (ns) of the circuit with 100 latches.

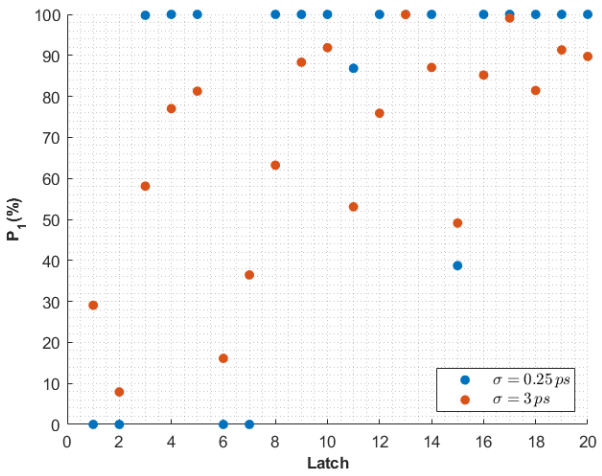

Fig. 13. The randomness of all 20 SR-latch TRNGs under different noise levels ($\sigma = 0.25$ ps and $\sigma = 3$ ps).

Equation 5 and Fig. 4 that we should get an entropy greater than 0.96 bit for $\sigma = 0.25$ ps (as its MNR is 8) and an entropy of 1 for $\sigma = 3$ ps. However, the Δ_M distribution is not centered and has a mean of $\mu = 5$ ps, which should deteriorate the entropy.

As expected and also shown in Fig. 13, we got the best randomness in our latches for the case of $\sigma = 3$ ps and the worst for $\sigma = 0.25$ ps. By the best we mean the majority of latches before being XORed have an acceptable randomness (distribution of '1' vs. '0') and by the worst we mean the circuit mainly behaves as a PUF (constant output). Hereby, we picked these two extreme cases for the aging analysis to analyze how device aging can affect the randomness of the TRNGs in these two cases. Please note that by randomness we mean the distribution of '1' and '0' values which is also referred to as uniformity in literature. More precisely, by randomness, we refer to the number of '1's generated by the TRNG over the total number of '1' and '0's.

D. Aging Induced Impacts on the SR-Latch Based TRNG's outcome

We used HSpice MOSRA to simulate the aging impacts on the targeted TRNG realized by 20 parallel latches for the course of 7 years with the steps of 6 months. The results are shown in Fig. 14 and Fig. 15. The former represents the best case mentioned earlier ($\sigma = 3$ ps) and the latter depicts the case in which ($\sigma = 0.25$ ps) which seemed to be the worst case among the ones considered in this study in terms of randomness. In each of these figures, for the sake of clarity, we only showed the randomness of two randomly selected (out of 20) latches, namely latches 11 and 15, along with the randomness of the whole TRNG realized by XORing the 20 latches. As shown, for the new device both latches (called Latch 11 and Latch 15) revealed highly acceptable randomness in the range of 50%±5%. The trend is the same during the course of aging for 7 years. Moreover, the main TRNG output realized by XORing all latches also exhibits a very similar (close to 50%) randomness. On the other hand as shown in Fig. 15, this trend is very different when the noise standard deviation (σ) is 0.25 ps. In this case, the selected latches exhibit very low randomness; having a randomness of 85% and 39% when new. This trend changes during the course of aging resulting in 29% and 66% randomness after 7 years.

Another important observation that can be made from Fig. 14 and Fig. 15 is that the randomness of the main TRNG (realized by XORing all 20 latches) becomes close to 50% in both cases of noise standard deviations equal to 3 ps and 0.25 ps during the course of usage even if for the latter the randomness was around 58% initially.

In sum based on the above discussion and observations we can conclude:

- Latch-based TRNGs may have a close to optimum behavior when the circuit timing (jitters) is accurately controlled. These optimizations might be achieved by explicitly using delay elements or an accurate placement and routing (PnR).
- On the other hand, for the circuit with a poor choice of standard deviations (noise not well-controlled), the value of randomness changes substantially. While the circuit gets close to the optimum point of randomness at some points (50%), not having a predictable behavior would give less confidence to the designer.
- Even for the circuit with poor control of randomness when new, one can achieve optimized randomness during the course of aging.

Figure 16 explains why the randomness of Latch 11 in Figure 15 varies substantially over time due to aging. As in this case (σ = 0.25 ps), the time difference between the falling edge of *S* and *R* signals is well below the Mismatch (Δ_M), and considering that the upper NOR gate operates much faster in the new device due to Mismatch (referring to the high value of the P_1 for Latch 11), the upper gate (generating the Q signal) gets evaluated before the other NOR gate. Under this condition (Q = 1 and $\overline{Q}$ = 0) that puts 0 to the input of the

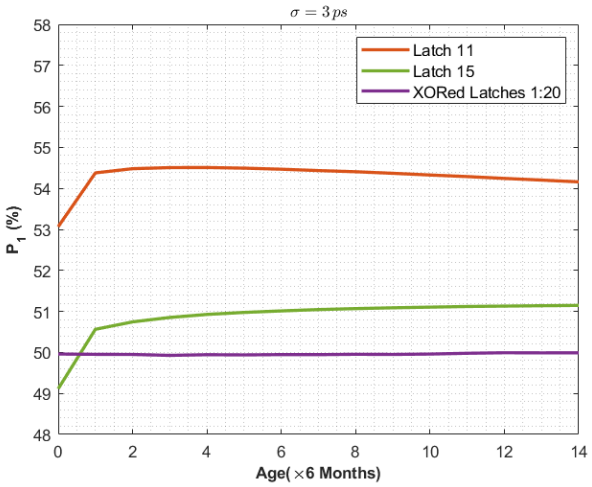

Fig. 14. Behavior of the circuit with noise standard deviation of 3 ps under 7 years of aging.

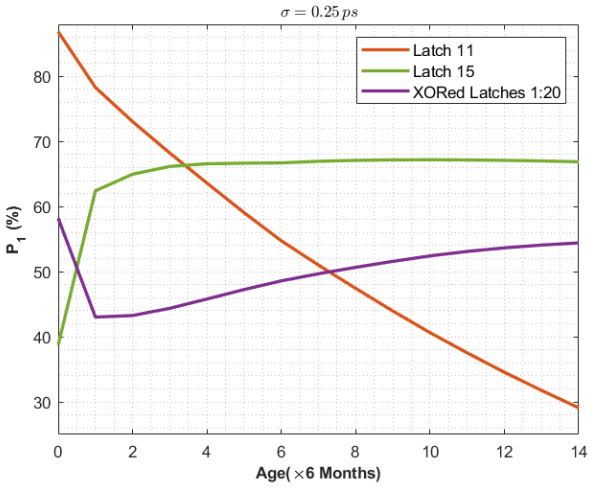

Fig. 15. Behavior of the circuit with noise standard deviation of 0.25 ps under 7 years of aging.

M3 transistor (it is assumed that these values are generated after the active edge of S and R signals goes from 1 to 0), PMOS transistor (M3) ages faster due to the NBTI impacts and the upper NOR gate gets slower while the other gate is not much stressed. It means the NOR gate at the bottom will be operating faster after some years than the upper one and $\overline{Q}$ path will be evaluated before the Q path resulting in more 0s at the output than 1s. Consequently, Latch 11 has a high P_i value in the beginning and gets very close to 30% at the end.

Figure 17 depicts the entropy value of the XORed 20 latches for two different value of $\sigma = 3$ ps, $\sigma = 0.25$ ps. As shown, the entropy for $\sigma = 3$ ps is stable at 0.5 for all ages of the circuit while this entropy for $\sigma = 0.25$ ps is increased at earlier ages of the circuit. This figure shows how a biased TRNG ($\sigma = 0.25$ ps) would reveal information to the potential attacker while the circuit with ($\sigma = 3$ ps) has good random

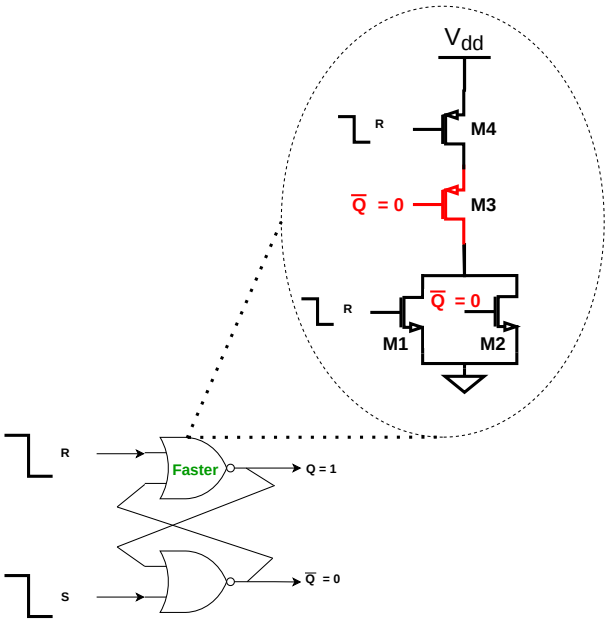

Fig. 16. Transistor level behavior of a faster NOR gate under aging.

TABLE I
NIST TEST RESULT RUNNING ON THE TRNG OUTPUT (AFTER XORING THE OUTPUT OF 20 INCLUDED LATCHES) BOTH FOR FRESH AND AGED CIRCUITS WITH $\sigma = 3$ PS.

Test	Age 0		Age 7-Year	
	Passed/ Total	P Value	Passed/ Total	P Value
Monobit	24/24	0.44	24/24	0.44
Frequency	24/24	0.47	24/24	0.45
Runs	24/24	0.50	24/24	0.54
Longest run	24/24	0.54	24/24	0.51
Binary matrix	6/8	0.51	7/8	0.39
DFT	23/24	0.44	23/24	0.42
Non overlapping	23/24	0.41	24/24	0.51
Overlapping	1/1	0.96	1/1	0.02
Universal	1/1	0.99	1/1	0.99
Linear Complexity	1/1	0.59	1/1	0.15
Serial	24/24	0.58	24/24	0.49
Approximate Entropy	24/24	0.62	24/24	0.51
Cumulative Sums	24/24	0.44	24/24	0.42
Random Excursion	1/1	0.42	1/1	0.43
Random Excursion Variant	1/1	0.45	1/1	0.43

characteristics both for fresh and aged devices.

Implemented TRNG (after XORing the 20 latches) was further evaluated using 15 statistical tests offered by NIST FIPS SP 800-22 [18] to assess their randomness within 1,200,000 cycles. The responses were divided into 24 blocks each including 50,000 random numbers, and we applied the NIST tests to each block. The results are shown in Table I and Table II when the circuit is new and aged for 7 years. Note that some of the tests (e.g., Universal) need larger blocks so we partitioned our responses accordingly. As shown in Table I, the TRNG passes almost all NIST tests when $\sigma = 3$ ps. On the other hand, when $\sigma = 0.25$ ps, the NIST tests are

TABLE II
NIST TEST RESULT RUNNING ON THE TRNG OUTPUT (AFTER XORING THE OUTPUT OF 20 INCLUDED LATCHES) BOTH FOR FRESH AND AGED CIRCUITS WITH σ = 0.25 PS

Test	Age 0		Age 7-Year	
	Passed/ Total	P Value	Passed/ Total	P Value
Monobit	0/24	0	0/24	0
Frequency	0/24	0	0/24	0
Runs	0/24	0	0/24	0
Longest run	21/24	0.235528	24/24	0.468735
Binary matrix	7/8	0.364195	8/8	0.354716
DFT	0/24	0.000349	21/24	0.323806
Non overlapping	5/24	0.114004	11/24	0.218996
Overlapping	0/1	0	0/1	0
Universal	1/1	0.952023	1/1	0.98463
Linear Complexity	1/1	0.638685	1/1	0.345916
Serial	0/24	0	0/24	0.139295
Approximate Entropy	0/24	0	0/24	0
Cumulative Sums	0/24	0	0/24	0
Random Excursion	1/1	0.491643	0/1	0.63267
Random Excursion Variant	1/1	0.596831	1/1	0.632342

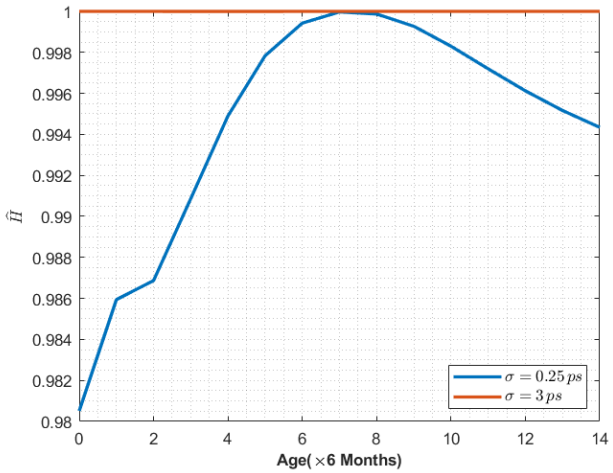

Fig. 17. Entropy of 20 latches (after XOR) for different σ values.

not successful (Table II). However, based on the information presented in these tables, it can be observed that the aging process not only does not lead to the degradation of the True Random Number Generator (TRNG) metrics, **but it also results in a minor improvement**, in some cases. For example, in Table I, the fresh device passes 6 out of 8 "Binary Matrix" tests while this number is 7 for the aged device.

In practice, it may not be possible to achieve the required randomness by only using a single latch as this mainly needs an analog/mixed-signal circuitry to change the jitter dynamically in order to achieve a randomness close 50%.

V. HEALTH TESTS IN VARIOUS CERTIFICATION SCHEMES (NIST SP 800 90B, AIS BSI 31, OSCCA GM/T 0078)

True Random Number Generators are amplifying some noise in order to create an unbiased and continuous flow of entropy. Even though the TRNG source has a sound design, multiple defects can cause it to malfunction. Let us list below a (non-exhaustive) list of reasons for a TRNG to fail:

- The TRNG can have a non-standard mode of operation, e.g., leveraging a free-running oscillator (FRO). The FRO is functioning at an impressive speed, and therefore likely cause local heat, which can result, in the long run, in some physical damage resulting from the thermal stress. In the case of a FRO-based TRNG, this can cause the loop to have such a breakdown that the loop gets opened. It is thus no longer oscillating hence no jitter (source of entropy) is accumulated. This dramatic situation is customarily denoted as "total failure".
- At a least extent, the TRNG source can become biased. Recall indeed that TRNGs are leveraging tiny noise and hence are fragile. For instance, the FRO can change speed, resulting in less entropy being collected, hence worth the quality of randomness when the circuit aged.
- Besides natural decay, a TRNG can also be manipulated by an attacker. Typically, the attacker would force the environment to couple to the FRO and therefore always resynchronize it, attenuating each effect of the jitter [3].
- Still more critically, the external source imposed by the attacker in the vicinity of the TRNG can be so strong that the FRO is completely controlled from the outside, yielding a mere cancellation of the noise due to jitter [13].

For all those reasons, it is reasonable not only to test the TRNG once at boot, but on a regular basis. A sane practice is to test the TRNG before each and every use. Such tests are termed "*health tests*" and are prescribed by multiple standards. Requirements for NIST, AIS, and OSCCA schemes are described next.

A. NIST SP 800 90B

The NIST is specifying in its publication FIPS 140-3 tests for cryptographic modules. An electronic device that is designed to pass all tests can be certified according to a FIPS 140-3 scheme, known as Cryptographic Module Validation Program (CMVP).

The 8th area of security requirements is termed "Sensitive Security Parameter Management", and includes the Random bit generators. In practice, the Entropy Source Validation (ESV) defers the evaluation to a Special Publication (SP) of the NIST, referred to as NIST SP 800 90B [21].

The ESV mandates independent health tests. They shall be run *at start-up* and *continuously*.

By default, two standard tests are indicated. Namely:

- Repetition Count Test (§4.4.1): it amounts to detect that some value is consecutively repeated many more times than expected, given the assessed entropy per sample of the source.
- Adaptive Proportion Test (§4.4.2): it aims at detecting that some value becomes much more common in the sequence of noise source outputs than expected, given the assessed entropy per sample of the source.

Those tests are reproduced in Alg. 1 and 2, where C is a given cutoff value. Obviously, some perfect sequences (meaning X_i are independent and identically distributed for all $0 \leq i < N$) will fail to pass the NIST canonical tests. Therefore the NIST prescribed that *false positives* occur with a limited rate between 2^{-20} and 2^{-40}.

Algorithm 1: NIST Repetition Count Test

static parameters : C: cutoff value.
input : Sequence of bits R_i ($0 \leq i < N$, where N can be equal to $+\infty$).
output : Failure or Success (or no response if $N = \infty$ and sequence X_i is random).

```
A ← R_0, B ← 1                                  // Initialization
for i ∈ {1, ..., N − 1} do                      // Continuous test
    X ← R_i
    if X = A then
        B ← B + 1
        if B ≥ C then
            return Failure
    else
        A ← X, B ← 1
return Success
```

Algorithm 2: NIST Adaptive Proportion Test

static parameters : C: cutoff value; W: window size.
input : Sequence of bits R_i ($0 \leq i < N$, where N is a multiple of W, and can be equal to $+\infty$).
output : Failure or Success (or no response if $N = \infty$ and sequence X_i is random).

```
i ← 0                                           // Initialization
for k ∈ {0, ..., N/W − 1} do                    // Continuous test
    A ← R_{Wk}, B ← 1
    for j ∈ {1, ..., W − 1} do
        if A = R_{Wk+j} then
            B ← B + 1
        if B ≥ C then
            return Failure
return Success
```

In order to deal with suspected failure modes that only the entropy source designers would be aware of, it is made possible by clause "Developer-Defined Alternatives to the Continuous Health Tests" (§4.5) to define custom tests. Those developer-defined continuous tests shall nevertheless detect failures timely.

The tests shall be run on at least 1,024 consecutive samples.

B. AIS BSI 31

The current applicable version is 2.0 [17]; notice that section §2.4 mentions other RNG standards. A new draft (version 2.35) is in preparation [9] and is open for public comments.

The requirements for PTG.2 and PTG.3 dictate a reliable online test (health testing) and a reliable total failure test that shall prevent undetected degradation of the entropy of the internal random numbers or an undetected total breakdown of the physical noise source.

The PTG.2 requirements (see §3.4.3 Functionality Class PTG.2) mostly focus on total failure tests and available entropy that can be extracted. There is currently no prescription regarding the health tests.

Notice however that the new version of the standard will require four novel online tests, namely:

- Test T1: monobit test
- Test T2: distance test
- Test T3: poker test
- Test T4: autocorrelation test on binary sequences

These will certainly have an impact on the complexity of health tests, since they are heterogeneous with other tests (such as that from NIST, namely Alg. 1 and 2). Thus a multi-compliance is still hard to achieve.

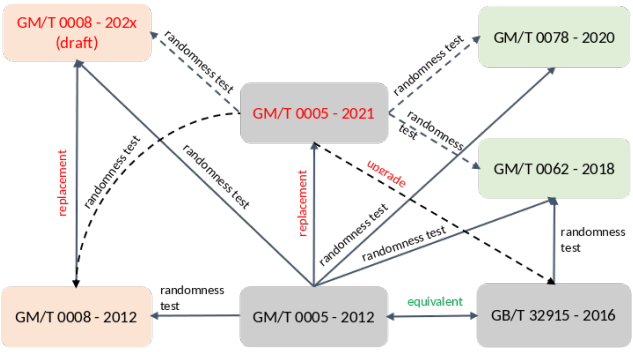

Fig. 18. Interplay between OSCCA standards for random number generation (Courtesy of Dr. Wei Cheng from Secure-IC certification team)

C. OSCCA GM/T 0078

The Office of State Commercial Cryptography Administration (abridged OSCCA) regulates the use and proper implementation of techniques involving cryptography in China. The OSCCA requirements regarding TRNGs have become stringent with the publication of GM/T 0078, which occurred in 2020 [5]. An overview of the OSCCA standards, already in place and to be launched, is given in Fig. 18.

An OSCCA-compliant module must implement four tests which are the same as that of NIST SP 800-22 T1, T2, T3, and T4, but on $2 \cdot 10^4$ bits. Notice that the test T3 of GM/T 0078-2020 (runs test) actually slightly differs from that of NIST SP 800-22. Besides, an OSCCA module must embed a total failure test, that reads 32 bits of data of the entropy source and returns an error defined as:

```
total_failure_error = (data == 0x00000000) \
                    | (data == 0xFFFFFFFF) ;
```

It is noteworthy that GM/T 0078 is dated from 2022, and then was **not** a requirement at the time GM/T 0008:2012 (or GM/T 0028:2017) came into force.

VI. Conclusions

We presented in this paper different methods to evaluate and validate a TRNG. We focused on the SR-latch TRNG which has a different principle than the well-known Ring-Oscillator TRNG. It is important, for instance in OSCCA certification, to implement jointly two TRNGs with different rationales, as they do not have the same failure modes, making the dual construction more resistant to attacks.

First, we propose a stochastic model of the SR-latch TRNG in order to validate its principle and adapt the architecture to the targeted entropy. It is notably shown that the number of latches of the architecture should be chosen according to the knowledge of process mismatch and noise. An ASIC implementation of 1024 SR-latches TRNG in 28 nm FD-SOI technology has shown the feasibility of the SR-latch TRNG to get a 1-bit entropy by using a subset of latches.

We moved one step forward and assessed the impact of aging on this TRNG type using HSpice/MOSRA. This study concluded that aging contributes to enhancing entropy if the mismatch to noise ratio is not too high. It also shows that the mean of the process mismatch should be relatively small compared to the noise level to keep a minimum number of latches near a metastable state, thus providing a high level of entropy.

Finally, a review of health tests is presented. It allows the user to verify that TRNG delivers a minimum of entropy and ensures no failure or attacks compromise the TRNG's ability to deliver sufficient entropy.

Acknowledgment

This research has been supported in part by the National Science Foundation CAREER Award (NSF CNS-1943224).

References

[1] "Nangate 45nm open cell library," "http://www.nangate.com".

[2] J. Bahrami, M. Ebrahimabadi, J.-L. Danger, S. Guilley, and N. Karimi, "Leakage Power Analysis in Different S-Box Masking Protection Schemes," in *2022 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2022, pp. 1263–1268.

[3] P. Bayon, L. Bossuet, A. Aubert, V. Fischer, F. Poucheret, B. Robisson, and P. Maurine, "Contactless electromagnetic active attack on ring oscillator based true random number generator," in *Proceedings of the Third international conference on Constructive Side-Channel Analysis and Secure Design*, ser. COSADE'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 151–166.

[4] M. Ben-Romdhane, T. Graba, and J.-L. Danger, "Stochastic model of a metastability-based true random number generator," in *Trust and Trustworthy Computing*. Springer Berlin Heidelberg, 2013, vol. 7904, pp. 92–105.

[5] Cryptography Industry Standard of the People's Republic of China, "GM/T 0078-2020: The Design Guidelines for Cryptographic Random Number Generation Module," December 28 2020.

[6] J.-L. Danger, S. Guilley, and P. Hoogvorst, "High speed true random number generator based on open loop structures in FPGAs," *Microelectronics journal*, vol. 40, no. 11, pp. 1650–1656, 2009.

[7] J.-L. Danger, R. Yashiro, T. Graba, Y. Mathieu, A. Si-Merabet, K. Sakiyama, N. Miura, M. Nagata, and S. Guilley, "Analysis of Mixed PUF-TRNG Circuit Based on SR-Latches in FD-SOI Technology," in *Euromicro Conference on Digital System Design (DSD)*, 2018, pp. 508–515.

[8] S. Kiamehr, M. S. Golanbari, and M. B. Tahoori, "Leveraging aging effect to improve SRAM-based true random number generators," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 882–885.

[9] W. Killmann and W. Schindler, "A Proposal for Functionality Classes for Random Number Generators," September 18 2011, Version 2.35 - DRAFT: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile&v=2.

[10] D. Kinniment and E. Chester, "Design of an on-chip random number generator using metastability," in *European Solid-State Circuits Conf.*, 2002, pp. 595–598.

[11] F. Lozach, M. Ben-Romdhane, T. Graba, and J.-L. Danger, "FPGA design of an open-loop true random number generator," in *Euromicro Conference on Digital System Design*, 2013, pp. 615–622.

[12] R. Maes, *Physically Unclonable Functions - Constructions, Properties and Applications*. Springer, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-41395-7

[13] A. T. Markettos and S. W. Moore, "The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators," in *Cryptographic Hardware and Embedded Systems (CHES)*, C. Clavier and K. Gaj, Eds., vol. 5747, 2009, pp. 317–331.

[14] M. Matsui, "Linear Cryptoanalysis Method for DES Cipher," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, T. Helleseth, Ed., vol. 765. Springer, 1993, pp. 386–397.

[15] A. Muthukumar, N. Sivasankari, and K. Rampriya, "Anti-aging true random number generator for secured database storage," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2017, pp. 1–7.

[16] F. Oboril and M. B. Tahoori, "Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level," in *IEEE DSN*, 2012, pp. 1–12.

[17] M. Peter and W. Schindler, "A Proposal for Functionality Classes for Random Number Generators," September 2 2022, Version 2.0 - https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Certification/Interpretations/AIS_31_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile&v=5.

[18] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications," NIST, april 2010, https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf.

[19] A. Schaub, J.-L. Danger, S. Guilley, and O. Rioul, "An Improved Analysis of Reliability and Entropy for Delay PUFs," in *Euromicro Conference on Digital System Design, (DSD)*, 2018, pp. 553–560.

[20] Synopsys, "HSPICE User Guide: Basic Simulation and Analysis," 2016.

[21] M. S. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, and M. Boyle, "SP 800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation," January 2018, https://doi.org/10.6028/NIST.SP.800-90B.