IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2025, No. 2, pp. 460–488. DOI:10.46586/tches.v2025.i2.460-488

Protection of Oscillator-Based PUFs against Side Channel Analyses by Random Interruption

Holger Mandry[1], Jonas Ruchti[2], Julius Nowakowski[1], Michael Pehl[2] and Maurits Ortmanns[1]

[1] Institute of Microelectronics, University of Ulm, Ulm, Germany, holger.mandry@uni-ulm.de
[2] Technical University of Munich, TUM School of Computation, Information and Technolgy, Munich, Germany, j.ruchti@tum.de

Abstract. Oscillation-based physical unclonable functions (PUFs) are known to be sensitive to power trace side channel analyses (SCAs). Although previous work investigated on countermeasures, these required significant additional amount of hardware or were just able to obscure sign information of a frequency comparison, while the magnitude information remains available to the attacker. As recent innovation on oscillation-based PUFs also require the magnitude-information beside the sign, e.g., to increase the reliability, the need arises to protect both. We present a new protection approach to hide both sign and magnitude information of oscillation-based PUF from an attacker. By introducing random interruptions in the oscillation, the power spectrum is blurred while the quality of the PUF is maintained. In addition to concept simulations and the discussion of different implementations, we use the example of a loop-PUF to show that the presented countermeasure can withstand several attack scenarios.

Keywords: ring-oscillator, physical unclonable function (PUF), loop PUF, side channel analysis

1 Introduction

As the world becomes more interconnected, the need to identify and protect electronic systems is critical. Physical unclonable functions (PUFs) have gained high interest in the last decade to detect and prevent counterfeiting and tampering, or as primitives for key storage and authentication applications. PUFs extract mismatches caused by manufacturing variations to derive random, but reproducible fingerprints of hardware devices. Since the information is extracted from unique unclonable physical parameters, PUF fingerprints can replace on-chip memory.

Oscillation-based PUFs are an exemplary realization and the main focus of this work. They derive their responses from comparing the frequencies of two nominally equal ring oscillators (ROs). While early work on PUFs focused mainly on single-bit responses, multi-valued (MV) readouts are attracting more and more interest. MV readouts can, e.g., be used to increase the extracted entropy per response or to increase the reliability.

As security primitives, PUFs have been subjected to many different attacks. For oscillation-based PUFs, side channel analyses (SCAs) observing their power consumption are a prominent approach to extract the secret information. While previous work has mostly investigated protection mechanisms for sign-based single-bit responses of PUFs [TDP20], the magnitude information used in MV readouts often remains available to the attacker.

Received: 2024-10-15 Accepted: 2024-12-15 Published: 2025-03-04

Contribution This work presents an *interruptible loop PUF architecture* that hides the actual oscillation frequency and thus the magnitude of an oscillating circuit from an attack via power trace analysis, such that applications using the PUF as secret information are protected from such SCA.

- To achieve this, we have developed a new and simple concept for tracking and holding an oscillation for dedicated small time steps. With a pseudo-random sequence of oscillation and interruption, the resulting power spectrum is blurred.
- We validate this concept through simulation and present several ways to implement the track-and-hold behavior in hardware with minimal additional components.
- With a hardware implementation the influence of the proposed concept on the PUF quality, e.g., bit-alias, uniformity, uniqueness and reliability is investigated and found to be in a negligible range.
- Power trace SCAs are performed on the implemented design and a state-of-the-art (SotA) loop PUF for comparison reasons. While the later one reveals the secret information, it can be shown that the interruptible loop PUF successfully hides the information.

Outline The paper is organized as follows: Section 2 provides background information about related topics. Section 3 introduces the new concept of an interruptible oscillation and proves the working principle by simulation. In Section 4 different hardware implementations are presented. The influence of the architectural changes on the PUF properties is examined in Section 5. SCA scenarios are performed and evaluated in Section 6. How the findings from the exemplary evaluated loop PUF can be applied to other oscillation-based PUFs is discussed in Section 7. Section 8 concludes the paper.

2 Background

2.1 Physical Unclonable Functions

PUFs extract hardware fingerprints from unique, unclonable manufacturing variations of nominally identical devices. In general, PUFs have an input vector, called challenge, which allows the selection of evaluated physical parameters. An extracted response together with its corresponding challenge is called a challenge-response pair (CRP).

There have been several proposals for PUF architectures in recent years, each of which has its advantages and disadvantages. In arbiter PUFs an impulse is triggered on two symmetrical paths. Due to process variations, the pulse travels faster along one path than the other. The resulting phase difference is detected by an arbiter and translated into a response bit [MV10]. The arbiter PUF has a short evaluation time but is sensitive to noise. By crossing the delay paths upon a challenge, multiple CRPs can be generated.

SRAM PUFs evaluate the meta-stable state of SRAM cells after start-up [MV10,HBF08]. SRAM PUFs are very area and power efficient, but in general have no challenge input.

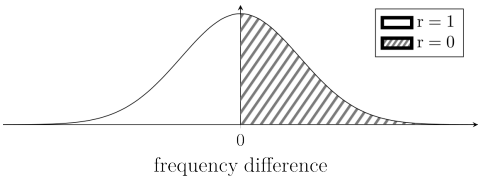
RO based PUFs rate the mismatch-dependent frequency difference of two nominally identical ROs [MCMS10]. By increasing the evaluation time, the noise affecting the RO PUFs can be reduced [HWGH18]. Due to their few design restrictions, RO PUFs are the preferred choice for field-programmable gate-array (FPGA) implementations [GCVDD02, CDGB12, MCMS10, HMBO19]. Variants of RO based PUFs have also been implemented in integrated circuits [DHWG20, PKS22].

2.2 Multi Valued Quantization

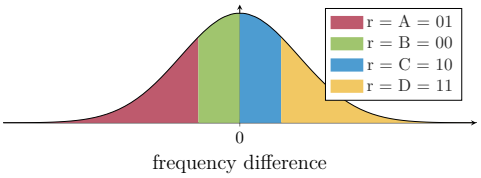
To increase the extracted entropy per response, several approaches have been investigated [IHL+19, MHM+20, CBD+18]. For example, in RO-based PUFs, it has been proposed to detect not only the sign of the frequency difference, but also its magnitude. By dividing the response space into subspaces, each representing one symbol as indicated in Fig. 1b, MV symbols can be extracted. As each symbol is represented by multiple bits, the extracted information per RO is increased ($\log_2 4 = 2$ bit in Fig. 1b).

Another potential application is to use the MV information to increase the reliability of the readout. The *two-metric helper data (TMHD) scheme*, first introduced by [DGS19] and improved in [TKDP21], is one exemplary approach for this. Its main idea is to increase the reliability of a *sign-based* decision with the help of the magnitude. Therefore the frequency range is split into four regions, each representing one combination of response bit and error-correcting reconstruction pattern (cf. Fig. 1c). In the initial phase, the reference value based on the sign and magnitude is determined and a helper data scheme for the reconstruction of later readouts is selected. The reconstruction scheme has slightly different areas to cover noise and therefore allows for a correct reconstruction. If, e.g., the response marked in red in Fig. 1c is extracted in the initial phase, the resulting bit is set to 0, and the blue reconstruction scheme is selected. If a re-extracted response is shifted to another region, e.g., by noise (red dotted in Fig. 1c), its magnitude is still mapped to the correct value with the help of the stored reconstruction scheme.

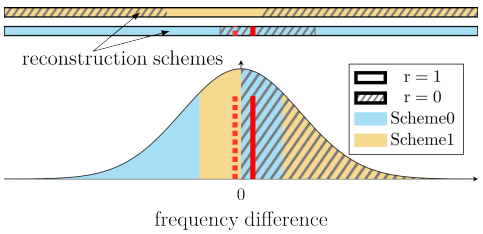
In [Fis24] another approach was presented, based on the idea of the MV decision combined with TMHD. There, the response space was divided into subspaces, each representing a symbol. Each subspace is alternately assigned to one of two reconstruction schemes. In the reconstruction phase, the helper data decides for the used reconstruction scheme, while sign and magnitude are mapped to the correct symbol.



(a) Sign-based decision, negative differences are mapped to a 1, positive ones to a 0



(b) MV decision [MHM+20], depending on the sign and magnitude the frequency difference is mapped to one of in this case four different response symbols



(c) TMHD decision as described by [TKDP21], the sign determines the response bit, while the magnitude is used to select a reconstruction scheme in the initial phase (red solid readout). A noisy readout with a potential sign flip (red dotted) is corrected with the help of the reconstruction scheme and the magnitude value.

Figure 1: Different quantization approaches to map frequency differences to bit values.

2.3 Typical Scenarios for Passive Attacks on PUFs

All *security-related applications* are exposed to attacks. Attackers have a wide range of options and approaches: machine learning attacks, which are not further investigated in this work, are used to predict the outcome of strong PUFs after training the neural network with a limited amount of captured CRPs [GTS15]. Introduction of non-linearity as with the XOR-PUF [SD07, GTS15] or specially designed PUFs [NSJ+18] are examples of protective countermeasures.

Decapsulation of the PUF chip allows access the secret information by probing the corresponding wires. Covering critical wires and nodes with dedicated potential planes is a countermeasure against such invasive attacks [ZHW+21]. Semi-invasive electromagnetic [MSSS11] and photonic emission [TNH+14, TDF+17] SCAs can be performed. However, as they require access to the backside of the chip and a more complex setup, we did not consider them further in this work and limited ourselves to non-invasive SCAs.

With *SCAs*, the frequency of an oscillating circuit can be observed from the outside. The switching operations of the oscillation temporarily increase the energy consumption at the same frequency. The highest power consumption (and thus the most leakage) is with the counter, where multiple registers need to be recharged [TPI19]. By Fourier-transforming the recorded power consumption in the time domain, the attacker obtains the power spectra, where spikes reveal the frequencies of the oscillations that occur in the system. If the attacker is able to identify the frequencies used to derive a response, it is possible for them to calculate the difference and thus reconstruct the response. Even though parallel oscillations, noise, and system clocks increase the difficulty of assigning the peaks with the response bits, they can still be obtained [MSSS11].

In [TDP20] the sign of a frequency difference was hidden from an attacker by randomizing the readout of the compared frequencies, and a sign-based decision as in Fig. 1a was protected. Nevertheless, the magnitude of the oscillation frequency difference remains inferrable to the attacker. For a MV readout PUF this reduces the possible symbols to two, while in the TMHD approach the response can be recovered with the stored helper data. Therefore, a protection mechanism to hide the frequency peaks in the spectra is desirable.

2.4 Recapitulation of the Loop PUF

The loop PUF, first introduced in [CDGB12], is based on the RO PUF, as shown in Fig. 2. Each delay element of the RO is connected to a challenge bit which allows to choose between two nominally identical delay elements that differ because of manufacturing variations. Each challenge evaluates a different combination of delay elements, resulting in a slightly different oscillation frequency measured by a counter, see Fig. 2.

By sequentially applying two different challenges and taking the sign of the difference of the resulting frequencies, a single bit is extracted as PUF response. In contrast to RO PUFs, which consist of two competing oscillators, this readout scheme has the advantage of avoiding deterministic frequency offsets due to the spatial distance of two individual ROs. To minimize the correlation between the applied challenges, Hadamard code words can be used, as they are linearly independent subset of all possible challenges, and thus prevent exploitable correlation of the extracted responses [TDP20]. To remove the bias, a challenge/Hadamard codeword and its bitwise inverse are used to derive a response.

2.5 Existing Loop PUF SCA Countermeasures

In [TDP20] a power-SCA was demonstrated. By analyzing the power supply spectrum during operation, it was possible to derive the frequencies of the oscillations for two evaluated challenges, which are compared to derive a single response bit. As the challenges

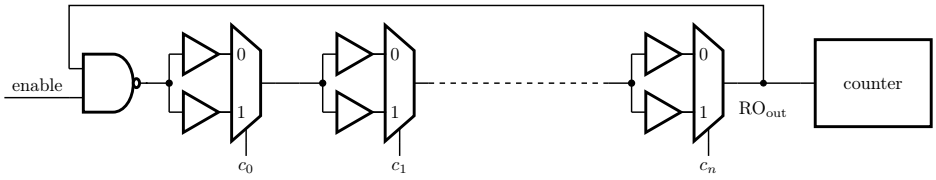


Figure 2: Implementation of a loop PUF with n delay elements, as introduced in [CDGB12]. While the enable signal is high the oscillation frequency, depending on the delays selected by the challenge input $c = [c_0 \cdots c_n]$, is evaluated by a counter.

are applied sequentially, it is possible to determine whether the first or second frequency is larger, which corresponds to the resulting PUF response. A countermeasure concept for this SCA was introduced in the same work: By randomizing the order in which the challenges are evaluated the attacker loses the information in which order the frequencies are subtracted. Therefore the sign of the difference, i.e., the PUF response, is hidden from the attacker. The randomness to permute the order of challenges is derived from the noisy counter LSB of the evaluation of a previous challenge for efficiency.

Although the sign of this oscillation comparison is hidden from the attacker when applying this random swap, the magnitude of the frequency difference is still known to the attacker ($|a - b| = |b - a|$). This is disadvantageous when MV-readouts are employed, as not only the sign but also the magnitude of the oscillation frequency carries a part of the secret information.

This was already identified as a potential weakness in [TDP20] and a further countermeasure was mentioned: since SCA require a minimum amount of time to analyze the power spectrum with meaningful resolution, it was suggested that the evaluation time of the loop PUF could be reduced to make the analysis more difficult. Disadvantageously, short evaluation times increase the error rate [HMBO19, HWGH18] and thus the reliability of the PUF response. This trade-off is undesirable. Therefore, an architecture that combines short oscillation periods to make attacks more difficult with long evaluation times to increase stability would be required.

Alternative approaches have been taken in [TKDP21] and [TDP22]: [TKDP21] proposes and analyzes randomization of the complete sequence of challenges, which solves the issue but requires significant hardware overhead. The interleaved challenge loop PUF in [TDP22] hides the response by modification of the PUF. It measures the two challenges being compared at the same time. In addition, it requires the measurement of the delay of the oscillation periods in the time domain, rather than the frequency, and uses an up-down counter for this purpose, which cannot be easily replaced. Not only is measurement in the time domain with a frequently switching up-down counter difficult to design and requires several synchronisers, but [SP24] has shown that the up-down counter requires a complex and potentially costly – in terms of area – design to avoid becoming the target of a SCA revealing secret information.

Therefore, we propose a design that also hides the response, but is easier to construct using frequency measurements, i.e. a simple counter that counts oscillations for a fixed period of time.

3 Interruptible Loop PUF to Avoid Attackability

In the following section, a countermeasure concept to protect the magnitude information of oscillating PUFs against SCA is proposed and verified by simulations. The *loop PUF* is chosen as exemplary implementation, but the concept can be extended to other oscillating PUFs.

3.1 Proposed Pseudo-Randomly Interruptible Loop PUF

Consider the following thought experiment: Imagine a trace of rectangular waves of an oscillator visible in the time domain for a fixed evaluation time. Slice the oscillation into time intervals shorter than one oscillation period. Define an operation that retains the logical value visible at some point in time for the length of one slice. Conceptually, similar to the insertion of dummy operations in a hiding countermeasure for AES, randomly insert such windows into the original oscillator trace. As a consequence, every period will have a different, unpredictable length, i.e., the frequency changes permanently and becomes blurred in the spectrum. This experiment describes on a high level the concept we introduce in the following, where the extending of the randomly inserted window is realized by interrupting the oscillator while keeping the phase information. From this fundamental concept, a number of essential requirements can be derived.

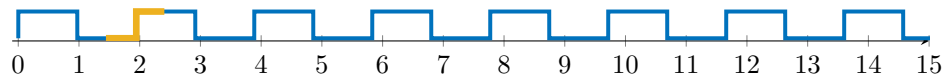
- First, it has to be possible to store/hold the current internal state (i.e., the current phase) of the oscillation during the interruption. Just enabling and disabling the RO will not fulfill this as the currently running oscillation would propagate its phase always to the interrupted feedback path, thereby discarding every frequency difference caused by mismatch.

- The duration of the on-phases must also be short, otherwise an attacker may still be able to measure the frequency.

- The sequence of on- and off-intervals has to be randomly distributed, otherwise it is possible to determine the on-phases and reconstruct the signal.

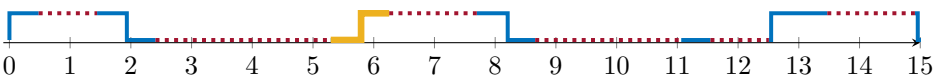
Fig. 3 illustrates an exemplary interrupted oscillation in the time domain. Note that the resulting interrupted RO signal is not just a simple multiplication of the RO and interrupt signal but an addition of time delayed RO cutouts (blue) together with sample and hold values (red dots). Also note that the sample and hold behavior causes a time shift of parts of the oscillation. This is highlighted in yellow in Fig. 3.

We call this proposed concept an *interruptible RO*. With an additional counter at its output, the amount of oscillations can be measured in a defined evaluation time, resulting in an *interruptible loop or RO PUF* architecture.



(a) Non interrupted RO signal

(b) Random interrupt signal



(c) Interrupted RO signal, where the dotted parts mark the interruption

Figure 3: Time signals of an RO, the interrupt signal, and the resulting interrupted RO in multiples of the interrupt clock cycle. The time shift of the oscillation due to the interrupt is indicated in yellow.

3.2 Concept Validation by Simulation

For validation of the proposed concept, an intuitive simulation of the *interruptible RO* was performed. In addition, a mathematical analysis of the countermeasure is given in Appendix A. For the simulation the output signal of an RO (RO_{out} in Fig. 2) with frequency f_{ro} was created in the time domain (cf. Fig. 3a), and interrupted by a pseudo-random pattern, clocked with $f_{int} = 2 \cdot f_{ro}$ (cf. Fig. 3c). The factor of 2 was chosen so that the interrupt signal is never slower than the oscillating RO by a sufficient margin.

In [TDP20], an oscilloscope with 250 MHz bandwidth was used to record the power spectra. This was 16 times larger than the actual RO frequency of 15 MHz on the used FPGA. In order to imitate this experiment in the simulations, a low-pass filter with a cutoff frequency of $16 \cdot f_{ro}$ was used to include the limits of measuring devices and prevent aliasing. The 100 ms long time signal, with simulation steps of 0.1 µs, of the continuously running RO and the interrupted RO were Fourier-transformed using the Matlab FFT function, and the resulting power-spectral-density (PSD) are plotted in Fig. 4: the continuously operating RO (in black) shows peaks at the oscillation frequency and its harmonics, with the drop at higher frequency bands coming from the applied LP filter. In contrast, the interrupted RO has a much more evenly distributed spectrum, with the highest peaks reduced by about 50 dB compared to the continuously operating RO, which already confirms the concept of the *interruptible RO* operation.

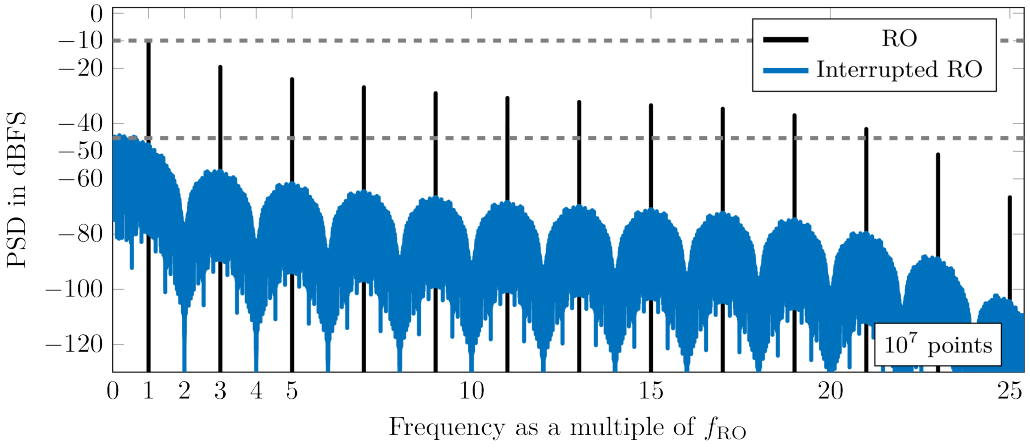


Figure 4: PSD of an RO in normal operation (black) and with random interruption pattern (blue).

Besides, the spectrum of the interrupted RO features notches at multiples of $2 \cdot f_{ro} = f_{int}$, which one might wonder if they reveal the oscillation frequency. This doubt is easily disproved: as the random interrupt signal is effectively a discrete-time (DT) noise sampled at f_{int}, it shows periodicity with f_{int}. After reconstruction with a (rectangular) zero-order hold (ZOH)-pulse, this introduces a sinc-filtering. Together with the introduced LP filter to mimic the limited bandwidth of an actual measurement of an oscilloscope, the resulting shape is understood. In other words the interrupt signal is a sum of rectangular signals of width $W = P \cdot T_{int}$, where P is a random number depending on the output of the pseudo-random number generator (PRNG) and T_{int} is the time period of the interrupt clock. These correlate to sinc functions in the PSD with zeros at multiples of $1/P \cdot f_{int}$. Since all of the sinc functions have a zero at $1/1 \cdot f_{int}$, the notches are formed. Since we have chosen the interrupt signal to be twice the oscillation frequency, Fig. 4 shows a correlation between both. In a real PUF, the individual oscillation frequency will be

randomly distributed around a mean frequency, which is the actual secret information. Then, the distinct notches in the spectrum of the interrupted RO are (mostly) related to the interrupt frequency, but not to the secret frequency of the evaluated actual RO. This is discussed in more detail below.

3.2.1 Variation of the RO Frequency

Up to this point, an ideal RO was simulated with an interrupt signal that was exactly twice as fast as the RO. However, the frequency of all ROs in a PUF will vary slightly from their nominal frequency due to manufacturing variations. To model this behavior in the simulation, the frequency of the RO was varied while the interrupt frequency was kept constant at $2 \cdot f_{\mathrm{ro}}$. In the following simulation, we applied a variation to two further simulated ROs named $\mathrm{RO_{L/H}}$. $\mathrm{RO_L}$ having an large frequency variation of $f_{\mathrm{roL}} = 0.97 \cdot f_{\mathrm{ro}}$, while $\mathrm{RO_H}$ is set to $f_{\mathrm{roH}} = 1.01 \cdot f_{\mathrm{ro}}$ which is in the reported range of variation for FPGA-ROs [HMBO19, HWGH18]. The spectra of all three continuous and interrupted RO output signals are plotted in Fig. 5.

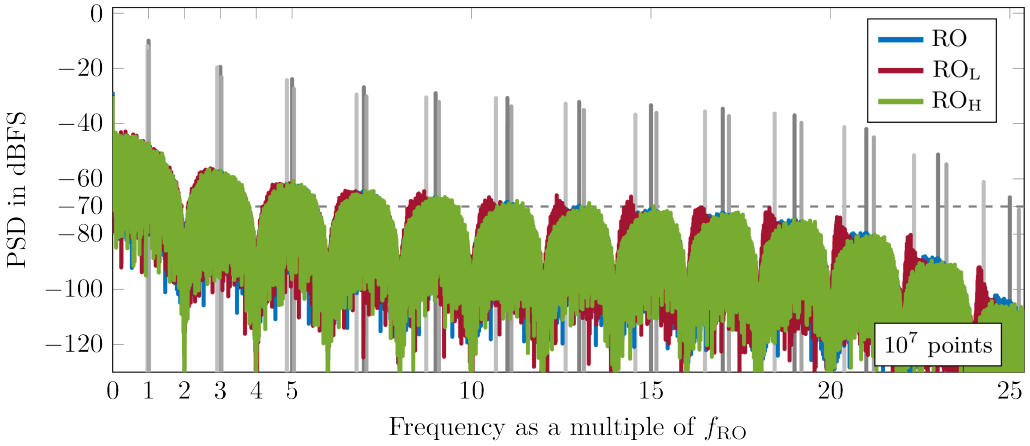


Figure 5: Comparison of ROs with different frequencies to show the influence of frequency variation on the resulting spectrum. The frequency of the interrupt was kept constant for all spectra. The non-interrupted frequencies and harmonics are plotted in gray.

Firstly, it is important to note that the distinct frequency peaks of the continuously operating RO, plotted in grey, reveal the secret information of slightly shifted oscillation frequencies. On the other hand, the interrupted RO successfully spreads the spectrum and the maximum power spectral density is reduced by more than 50dB! Though, a closer look reveals some potentially remaining secrecy leakage: while for low frequencies all interrupted spectra look almost identical, towards the higher bands, the spectrum of $\mathrm{RO_L}$, which has a lower frequency than the original RO, starts sloping to the left, while the spectrum of $\mathrm{RO_H}$, which is a bit faster than the original RO, starts sloping to the right. As all of them were simulated with an identical interrupt pattern this must be related to the secret frequency information.

In Section 6.1 this vulnerability is tested in practice. The underlying effect is understood as follows: while we explained earlier why a sampled random signal together with a rectangular ZOH has a sinc-shape, cf. Fig. 4, multiplying it by a unique RO spectrum shifts the lobes according to the position of the RO frequency. This shift is relatively constant – determined by the percentage shift of an actual RO frequency – but becomes absolutely larger at the higher harmonics of the oscillation frequency. Therefore, the shift

becomes larger towards higher frequencies in Fig. 5, and it is more prominent for the rather large shift of f_{roL} vs. the smaller shift of f_{roH}.

On the plus side, these shifted lobes have >60 dB lower power spectral density than the main frequency peak of the continuously operating RO. Although the direction and gradient of the slope can give an indication of the ratio of the observed frequency to the ideal/mean frequency f_{ro}, extracting its magnitude is therefore orders of magnitude more difficult. Moreover, the absence of noise in the shown simulation makes this effect visible in Fig. 5, while in a real measurement noise would blur the spectrum and this secrecy leakage even more. The same blurring would result from actual non-rectangular pulses due to finite rise and fall times in a real implementation. In order to obtain realistic noise parameters, measurements must be carried out to obtain these values. We have therefore decided not to include noise in the simulation and instead show the measured spectra in Section 6.

3.2.2 Variation of the Pseudo-Random Number Generator Seed

Another parameter that can be varied is the PRNG sequence. Ideally, the ratio of ones and zeros should be the same for all seeds for the PRNG sequence, resulting in the same off-phase duration. However, as the runtime (and simulation time) is limited, there are some differences in the interrupt pattern and hence the resulting response: due to different amount of on-phases with variable length per pattern, the total amount of rising edges due to the oscillation will vary (quantization error). Consequently, both ROs used in an actual comparison to generate a PUF bit must be interrupted with the same seed, i.e., the same interrupt pattern. With a simulation, we have verified that the resulting error is within an acceptable range and comparable to common error rates due to noise [HMBO19, HWGH18].

Furthermore, differences in the resulting spectra can be expected because short sequences of alternating ones and zeros have higher frequency parts than longer sequences, which should be reflected in the spectra. Therefore, $\mathrm{RO_L}$, which showed the most noticeable shift of the higher frequency lobes in Fig. 5, was re-simulated with 100 different random seeds for the PRNG. In Fig. 6 the spectra of $\mathrm{RO_L}$ are shown for three exemplary different random seeds. They look almost identical with only very minor differences, thus we can conclude that the spectrum is not affected by the seed.

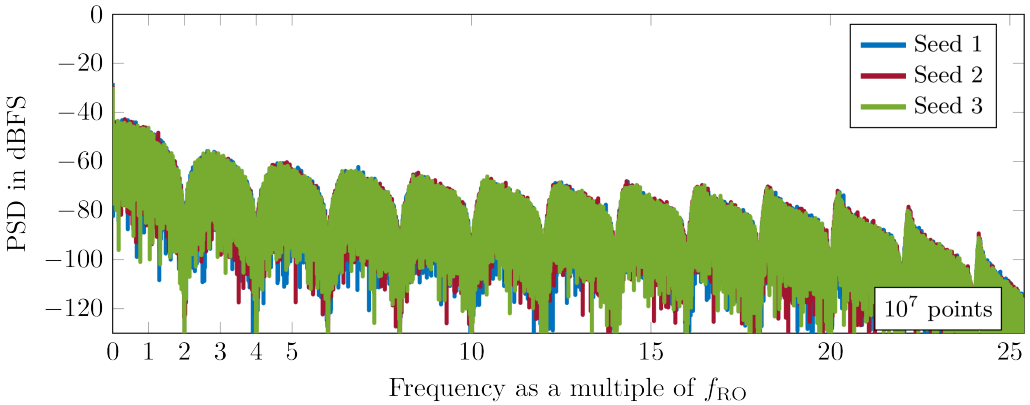


Figure 6: Spectra of $\mathrm{RO_L}$ interrupted with three different seeds for the PRNG.

In summary, the shown simulations prove that an *interrupted RO* features a blurred power spectrum, where the maxima in the PSD are reduced by about 50 dB, while the potentially exploitable information leakage is even reduced by more than 60 dB in the shown ideal simulations. Including non-idealities in both the ROs and the readout, the

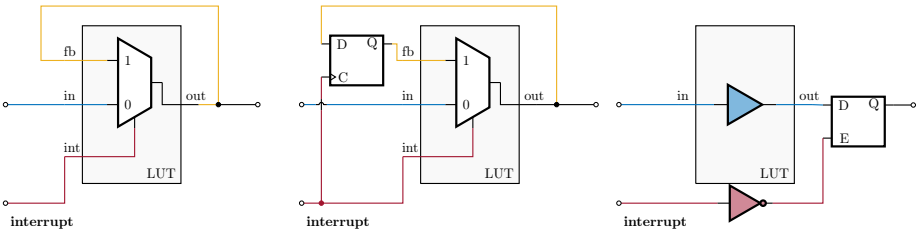
reduction in information leakage is expected to be even higher. Moreover, in order to retrieve the correct PUF response, the compared RO should be interrupted with the same pattern. These findings motivate the implementation of the proposed technique in an actual hardware to enable the validation of its behavior by measurements.

4 Implementation

The proposed *interruptible RO* is implemented on an FPGA since this allows easy reprogramming and is well suited for prototyping and concept evaluation. Next, the possible implementations of a suitable delay element realizing the interrupt feature are discussed. Thereafter, the overall interruptible RO PUF is introduced.

4.1 Implementation of a Delay Element With Track and Hold Function

The concept of the interruptible RO requires to store the actual (phase) state of a running oscillation to maintain the PUF functionality, cf. Section 3.1. Therefore, each delay element is extended by an additional *interrupt* input. If the *interrupt* input is logical low, the normal delay operation is performed. Once *interrupt* becomes high, the current output value must be held constant regardless of the signal input. On FPGAs the delay elements are usually realized by lookup-tables (LUTs), thus we will use it as a starting point for possible extensions to realize the required track-and-hold function within the delay elements, which will be explained in the following.



(a) Latching LUT **(b)** Additional Flip-Flop **(c)** Additional D-Latch

Figure 7: Different delay element implementation approaches to sample and hold the input signal (in blue), while the interrupt (red) is high. (a) and (b) make use of an additional feedback path (yellow), while (c) adds a latch into the signal path.

Latching LUT

A first implementation variant is to feedback the output of the LUT to one of its inputs (cf. Fig. 7a). As long as the *interrupt* is low, the signal input *in* is propagated to the output *out*. When the interrupt changes to high, the signal input is ignored and the input 'fb', which is equal to the output signal, is propagated to the output; this forms a latching LUT. This type of implementation requires one additional wire as feedback. A disadvantage of this implementation is that propagation delays in the feedback result in race conditions if the interrupt and signal input change simultaneously. In the worst case, this could lead to internal oscillations on the feedback path within the LUT, causing one or more glitches.

Additional Flip-Flop

An additional flip-flop (FF) in the feedback path can prevent continuous internal oscillations. By connecting the interrupt input to the clock input of a FF, the current output value of the LUT is sampled when the interrupt rises, and it will be fed back as long as the interrupt stays high (cf. Fig. 7b). As the output is only sampled on the rising edge, internal oscillation is prevented while normal operation is unaffected. However, different propagation delays on the interrupt line can cause glitches, potentially increasing the error rate, if the sampled value of the last interrupt is propagated to the output before the sampled value is updated.

Additional D-Latch

To eliminate the feedback loop in each delay element as required in Fig. 7a and Fig. 7b, alternatively a latch (LDCE primitive [Xil13]) can be included into the forward path as shown in Fig. 7c. As long as the interrupt signal is low, the latch is transparent for the oscillation signal, but blocks the transmission as long as the interrupt is high. As the LDCE is transparent for a logic high signal, an inversion of the interrupt is necessary. This approach eliminates race conditions, but also affects (reduces) the RO frequency, as the forward latches introduce additional delay into the data path. This will affect the required evaluation time for a given number of oscillations.

4.2 Phase Quantization Errors

In addition to possible glitches, the proposed implementations introduce other sources of potential errors, which are mentioned in the following.

For loop and RO PUFs, the secret information is derived from the phase difference of two oscillation frequencies integrated over the evaluation time. Thus the quality of the response depends on the accuracy of the phase difference and the evaluation time. While the oscillation is stopped and the state is sampled by the storage elements of the delay stages, it is still possible that a signal change propagates till the next storage element, adding a quantization error to the phase difference that depends on the delay between two subsequent storage elements. For example, suppose the signal change of the oscillation is "located on the wire" between two storage elements when the interrupt is triggered, it will propagate to the next delay element. Considering the oscillation it is compared to is sampled without error, the signal catches up or runs away from this oscillation. The resulting maximum quantization error in a loop per interrupt is the delay between two storage elements.

Furthermore, the signal changes have a limited slope, which means that analog voltages occur in the circuit. In memory devices such as FFs or latches, the hold functionality is realized by two coupled inverters forming kind of an SRAM cell. Metastable states arise when a cell is initialized with a value between VDD and GND, which occurs when the input signal changes while the interrupt is triggered; this is usually addressed by respecting setup and hold times in digital design. Depending on the initial value and manufacturing variations, the SRAM will decide on an unpredictable but deterministic (assuming no noise) value. If the metastable state resolves to the opposite of the previous output, the signal change can propagate to the next element, adding a phase error. In the worst case the SRAM cell will oscillate several times until it settles. If this happens in the element connected to the counter, the counter value may be erroneously incremented, depending on the internal oscillation frequency and the setup and hold time requirements of the counter.

4.3 System Implementation

For each type of delay element, a separately placed and routed design has been implemented on an FPGA, each with a total of 72 loops and a challenge width of 16 bit, and a bitstream was generated. Each loop is connected to a 32 bit counter and can be enabled individually. The random interrupt sequence is generated by a PRNG which is emulated by an linear feedback shift register (LFSR) clocked by a separately controllable system clock. The LFSR requires 38 LUTs and 72 FFs. In addition, a buffer is added for every four LUTs to reduce the fan-out of the interrupt signal. A built-in timer is used for the enable signal of the individual PUF cells.

For the FPGA implementation, the additional overhead of the proposed countermeasure is mainly limited to the LFSR. The overhead per implemented loop is one additional FF per delay element for the *D-latch* and *Flip-Flop* approaches, and none for the *Latching LUT* variant. However, these elements are still blocked in the unprotected version to minimize crosstalk. Similar to [TDP20], the response of a loop with applied challenge c is calculated off-chip with

$$r(c) = \text{sign}(\text{CNT}(c) - \text{CNT}(\overline{c})) \tag{1}$$

with $\text{CNT}(c)$ being the counter value after the evaluation of challenge c and $\overline{c}$ being the bit-wise inverted challenge. Switching c to $\overline{c}$ changes the internal signal path of the LUT and thus its propagation delay, which is a common practice [TDP20]. 16 bit Hadamard codewords are used as challenge input, to prevent exploitable correlation of the responses, as stated in Section 2.4. To avoid bias, the all-zero code word is omitted as its Hamming weight differs from all other code words [CDGB12, TDP20], resulting in a total of 15 different responses per loop. The seed of the PRNG is kept constant for one response, so that the challenge and its inverse are interrupted with the same random pattern, to limit the quantization error on the response (cf. Section 3.2). For subsequent challenges, a new random seed is selected.

5 Evaluation of the PUF Quality

Even though the intention of the work is to introduce random interruptions of oscillator based PUFs and enhance resilience against SCA, the first evaluation of the proposed *interruptible RO PUF* is on its PUF quality metrics. As the aim of this work was not to design a PUF with more ideal characteristics compared to the SotA, we focus on showing that our modification of an existing PUF to increase the resistance against SCA has little/no effect on the PUF characteristics. To be comparable with other SotA publications, we choose the common metrics reliability, bit-alias, uniformity and uniqueness for this purpose [MCMS10]. Besides the classical *sign-based* bit quantization, the *TMHD* method is evaluated in this regard as a representative of approaches utilizing the magnitude of RO frequencies, since protecting this magnitude against SCA is the main motivation of the interruptible RO PUF. TMHD was chosen as it utilizes the magnitude for stabilization but results in a one-bit response, which simplifies the comparison with the sign-based approach.

5.1 Qualitative Evaluation of Different Delay Elements

The first experiment investigates the influence of the proposed interruptible delay elements as described in Section 4.1. To rate the reliability of PUFs the average bit error rate (BER) is often used in the SotA [MCMS10]. It is calculated by

$$\text{BER} = \frac{1}{R-1} \sum_{i=1}^{R-1} r_i \oplus r_0 \cdot 100\,\%, \tag{2}$$

Table 1: Measured mean frequency and mean bit error rate for different implementation variants of the interruptible delay element for continuous and interrupted operation. Results from a single FPGA with 72 16 bit-loops and 100 repeated read-outs for each challenge. Bit error rates use sign-based quantization. The oscillation frequency is calculated on counted rising edges at the RO output divided by the constant evaluation time.

	Continuous		Interrupted	
Variant	Frequency (MHz)	BER (%)	Frequency (MHz)	BER (%)
Latching LUT	49.96	4.1	24.90	13.9
Flip-Flop	48.58	5.1	34.51	13.1
D-Latch	38.44	4.1	19.54	5.3

where the first sample r_0 of R repetitive measurements is taken as reference response. As a second criterion, the oscillation frequency was calculated by the number of counted rising transitions at the output of the RO divided by the constant evaluation time of 10.486 ms. Table 1 reports the mean results for the BER and oscillation frequencies of all 72 implemented loops of one FPGA based on 100 readouts per challenge pair for the three different delay elements, both for inactive and active interrupt. Note that although the number of 100 readouts for stability analysis may seem low, it is sufficient for an initial assessment of the architectures.

The results confirm that the mean frequency of the implementation using the forward latch implementation (cf. Fig. 7c) is reduced from about 50 MHz to about 38 MHz compared to the other implementation approaches, which was expected due to the additional elements in the signal path. We can also see that in continuous operation (without interrupt signal) the BER remains almost constant for the evaluated approaches. Under interrupted operation and for unchanged evaluation time, we expect the measured mean frequency to drop by about 50 % since the PRNG with equal probability of 1 and 0 activates the oscillation for about 50 %. This is confirmed for the latching LUT and the D-latch.

By contrast, the measured mean "frequency" for the flip-flop extended version of the interruptible delay element is significantly higher at about 71 % of the continuously operating RO. Further investigation of the implemented design revealed more delay from the interrupt line to the clock input of the FF than to the multiplexing LUT due to asymmetric routing in Fig. 7b. We have verified by simulation that this causes deterministic glitches when the interrupt rises and the multiplexer forwards the old value of the FF until the FF is clocked, storing the current value. The probability that the old and current value differ is 50 %. Considering the deterministic frequency reduction to 50 % due to interrupted operation, the probability of an additional glitch per oscillation is about 50 %; this results in a total amount of transitions at the output of the RO at 75 % of the continuous operation, which is close to the measured value.

Concerning the BER for both the latching LUT and the FF-based delay element, an increase of about 2.5 to 3 times can be observed, which is unacceptable and underlines the quantization errors and setup and hold time violations described above. In contrast, the D-latch implementation only suffers an increased BER by about 1 percentage point, which seems acceptable. Since there is no local feedback (adding propagation delay) with the D-latch, the described metastable states can settle much faster than with the LUT or FF-based approaches.

As a result from this analysis, the D-latch implementation of the interruptible delay element is chosen for further analysis, as it is easy to implement and has the least change in BER during interrupt mode, cf. Table 1. The reduction of its mean frequency by about 20 % caused by the additional delay in the forward path leads to reduced effective

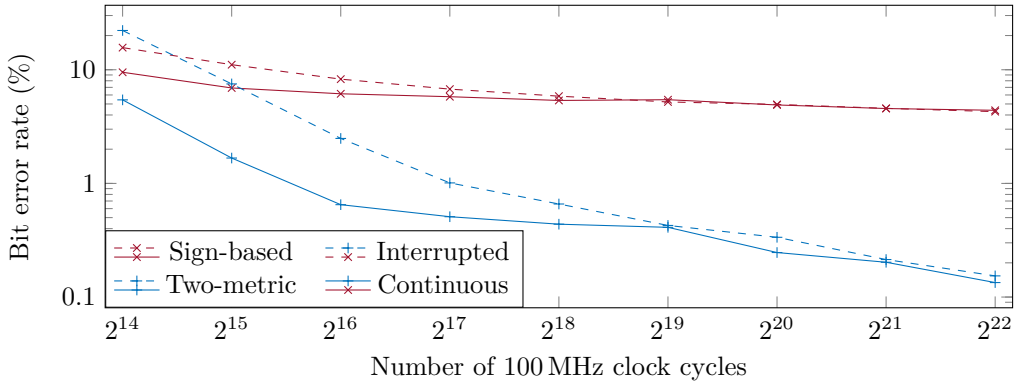


Figure 8: Dependency of the bit error rate on the loop runtime expressed as 100 MHz reference clock cycles and the total time for evaluating all 30 challenges of a loop. Continuous and interrupted modes using sign-based or two-metric quantization are displayed. Averages from 6 FPGA boards and 1001 repetitions were used.

evaluation time and thus reduced edge transitions and thereby has an impact on the BER; though, its BER is still by far the best among the alternatives in Table 1.

5.2 Influence of Evaluation Time

After the design of the delay elements is fixed, there are still free parameters to choose. The *total oscillation time* for the loops constitutes a general trade-off in RO-based PUFs: longer evaluation times allow to average more oscillation cycles and therefore enhance the reliability and BER. In practice the underlying application will limit the maximum latency before a PUF response must be available. Fig. 8 shows the result of an experiment with different loop runtimes. In this experiment, we employ the continuously running loop PUF and its proposed interruptible version, each using the latch-based delay element, cf.Table 1. The runtimes are shown as a number of clock cycles of a 100 MHz reference clock on a logarithmic scale. Results are plotted for both the sign-based evaluation as well as the TMHD evaluation.

As expected, the BER decreases as the reference count and thus the evaluation and oscillation time are increased. Moreover, the sign-based evaluation is always worse by almost an order of magnitude compared to the TMHD evaluation, which is expected as the main aim of the TMHD approach was to increase the readout reliability of a sign-based decision by incorporating the magnitude of the difference in oscillation frequency.

If we compare the continuously running loop PUF with its interrupted counterpart, we observe a slightly worse result up to a runtime of $2^{19} - 1$ reference clock cycles. This decrease is not only caused by halving the effective oscillation time due to the interruption during 50 % of the evaluation time, but also due to additional error sources discussed in Section 4.2. After this runtime, the BER for continuous and interrupted mode are almost identical, which is a known behavior, because after a certain point, increasing the evaluation time has no significant effect on the BER. [HWGH18]. Therefore we focus on $2^{19} - 1$ clock cycles as evaluation time for the final evaluation, for both the continuous and the interrupt mode. Hence the evaluation of a complete loop with all 30 challenges last 157.5 ms.

Since this is a quite long evaluation time, we also examined evaluations with fewer reference clock cycles. For the reference clock cycles $2^{14} - 1$ and $2^{15} - 1$ not only the BER is relatively high but also the bias and uniqueness are quite worse. For example

the bias of the sign-based readout drops to 0.4 in the continuous mode and 0.29 in the interrupted mode for $2^{14} - 1$ clock cycles. This is due to the fact that the loops oscillate relatively slowly. At least $2^{16} - 1$ clock cycles are needed to accumulate enough phase difference to get meaningful integer differences in the counters. This could be solved by implementing faster loops, but since the interrupt frequency needs to be faster than the oscillation frequency, and setting the interrupt frequency on the FPGA is restricted, we decided to keep the loops slow to be more flexible in changing the interrupt frequency. To evaluate more realistic readout times we additionally used $2^{16} - 1$ clock cycles for the final PUF quality evaluation, resulting in 19.6 ms to evaluate a loop with all challenges.

5.3 Influence of Interrupt Frequency

Since it has already been shown that the interrupt slightly worsens the BER, the question arises as to how the interrupt frequency affects this change. As a higher frequency results in more transitions between on- and off-phases, additional glitches and thus an increasing BER can be expected. To investigate this behavior, the interrupt frequency was swept from 1 to 5 times the approximated RO frequency $f_{\text{base}} = 40\,\text{MHz}$. This frequency differs slightly from the measured value of 38.44 MHz (cf. Table 1); the approximation was chosen to simplify the clock signal generation. The resulting BERs are plotted in Fig. 9 over varying interrupt frequencies. As expected, the BER increases with higher frequencies, but compared to the BER of 4.06 % of the continuous mode (cf. Table 1), the change is still smaller than 2 percentage points.

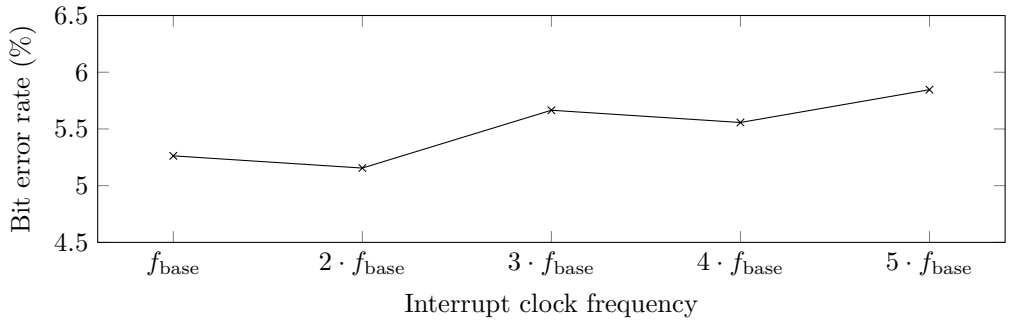


Figure 9: Bit error rate for sign-based quantization for different interrupt frequencies, expressed as multiples of the approximate continuous loop frequency $f_{\text{base}} = 40\,\text{MHz}$. 6 FPGA boards with 72 16 bit challenges and 101 evaluations at $2^{19} - 1$ (100 MHz) reference clock cycles were used.

5.4 Evaluation of Bit-Alias, Uniformity, Reliability and Uniqueness

After having chosen the interruptible delay element and the evaluation time, measurements to rate the PUF quality can be performed. A total amount of $M = 101$ Artix 7 FPGAs on Basys 3 boards, each implementing $L = 72$ loops, have been evaluated over $R = 1001$ repetitive readouts. Using the described Hadamard codewords as loop challenges, each loop generates 15 bit. Concatenating the response of 72 loops, each of which provides 15 bit information in a single row, results in a PUF secret S of B =1080 bit per FPGA board.

First, the *bit-alias* and *uniformity* [MCMS10] are evaluated, which both indicate the distribution of ones and zeros per bit position along all chips and per extracted secret, respectively. The bit-alias is calculated by $\frac{1}{M}\sum_{i=1}^{M} r_{i,t}$, where $r_{i,t}$ is the t-th position in S of FPGA i, and the uniformity via $\frac{1}{B}\sum_{t=1}^{B} r_t$ [MCMS10]. A value of 0.5 is the

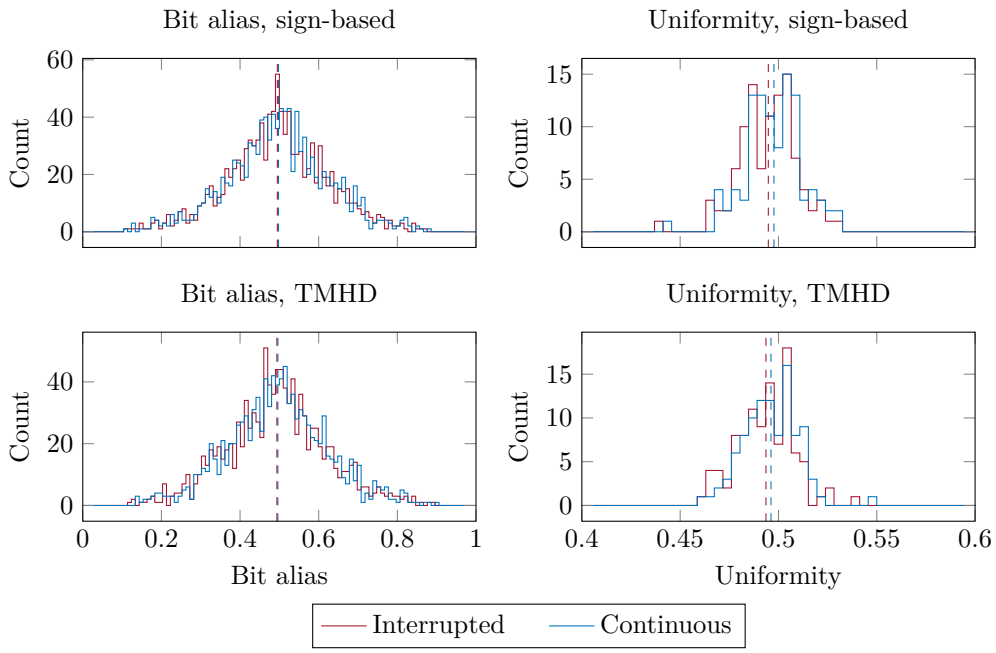


Figure 10: Comparison of bit-alias and uniformity, computed from 101 FPGA boards with 1080 bit responses each, for interrupted and continuous operating mode, separated by sign-based (top) and TMHD quantization (bottom). In addition, the means of the distributions are plotted. $2^{19} - 1$ clock cycles were used as evaluation time.

desired optimum, while outliers in the binomial distribution indicate a tendency of single bit-positions towards either constant zeros or ones.

The resulting distributions of the bit-alias and uniformity for continuous and interrupted modes are plotted in Fig. 10, separated for sign-based and TMHD approach. The distributions of the continuous and the interrupted mode are very similar showing that the interrupted mode adds no additional bias. All have a binomial-shaped distribution with a mean value close to the optimum of 0.5, which further indicates that the chosen design features no strong bias, neither for the normal operation nor the proposed interrupted mode.

Further, the *reliability and uniqueness* of the implemented loop PUF are evaluated. The uniqueness is calculated by

$$\text{uniqueness} = \frac{2}{M \cdot (M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \frac{\text{HD}(S_i, S_j)}{B}, \tag{3}$$

with HD being the Hamming distance of two secrets S of two different FPGA boards [MCMS10]. A mean value of 0.5 is desired for a good PUF.

Table 2 compares the results of the measurements for bias, uniqueness, and reliability for continuous operation and interrupted mode. With the fast evaluation time of $2^{16} - 1$ clock cycles the results of the continuous mode with sign-based quantization are not ideal but acceptable. For the interrupted mode, both the bias and the BER worsen noticeably. Since the interrupt halves the effective evaluation time, more loops do not perform enough oscillations to accumulate notable frequency differences (cf. Section 5.2), quantized by the counter, and noise becomes more dominant in the response.

With the longer evaluation time of $2^{19} - 1$ clock cycles, the quality is very good. While the uniqueness changes only insignificantly when the interrupt is turned on, the bias

worsens by about 0.3 %, which are negligible deviations. The reliability decreases by about 0.6 % for the sign-based approach and 0.14 % for TMHD, respectively. This is expected behavior, since the evaluation time is the same for continuous and interrupted mode, the pure oscillation time is halved with an active interrupt, which worsens the BER.

In absolute values, the bias is close to the optimal value of 0.5. The uniqueness is 2.3 %/1.1 % below the optimum, but still reaches an adequate value compared to other work [CDGB12, MBC16]. The BER can naturally not reach the theoretical optimum of 0 %. Though, by comparing the sign-based readout with the TMHD approach, this shows almost an order of magnitude better BER for both the continuous and the interrupted operation. This shows that the remaining BER is not a fundamental problem of the interrupted operation.

5.5 Reliability over Temperature and Voltage Variations

Since environmental changes are known to affect PUF responses, the influence of temperature and voltage variations on the reliability was also investigated. Three CW305-boards were used for this, as they have a controllable on-board voltage regulator that allows the supply voltage of the Artix 7 FPGA to be changed directly. All boards were placed in a temperature chamber to control the environmental temperature. In Order not to destroy the board components the temperature was swept from 0 °C to 75 °C in steps of 25 °C, while the voltage was set to 0.9 V, 1.0 V, and 1.1 V. In Fig. 11 the measured BER for sign-based and TMHD readout are plotted with $2^{16} - 1$ and $2^{19} - 1$ clock cycles as evaluation time. The normal operating voltage of 1.0 V and room temperature 25 °C are used as reference. For the continuous operation, plotted as continuous lines, the BER remains fairly constant over temperature variation, which is an expected behavior of the loop-PUF [TDP22], since the same hardware with different challenges is used to derive the response. Voltage variation, indicated by different markers, results in a small increase in BER, whereby the TMHD readout scheme is able to correct most of the errors as expected.

For the interrupted readout, plotted as dashed lines, the increase in BER is remarkably higher. For the sign-based readout it increases by about 4 percentage points for a temperature delta of 50 °C. This happens for several reasons. First, the interrupt clock is generated from the on-board PLL and thus not as temperature sensitive as the pure ROs. We already know from Section 5.3 that changes in the interrupt frequency compared to the oscillation frequency have an impact on the BER. However, since this influence was found to be quite small and the oscillation frequencies do not change by integer factors there must be an additional source of error, related to the interrupt.

Table 2: Metrics computed from 101 FPGA boards with 72 loops, 101 repetitions for $2^{16} - 1$ and $2^{19} - 1$ (100 MHz) reference clock cycles.

Metric	Continuous 2^{16}	Interrupted 2^{16}	Continuous 2^{19}	Interrupted 2^{19}
Sign-based quantization				
Bias	0.480	0.455	0.498	0.495
Uniqueness	0.475	0.472	0.472	0.473
Bit error rate (%)	6.146	8.267	4.623	5.250
Two-metric quantization				
Bias	0.469	0.443	0.496	0.494
Uniqueness	0.475	0.472	0.472	0.472
Bit error rate (%)	0.650	2.487	0.222	0.367

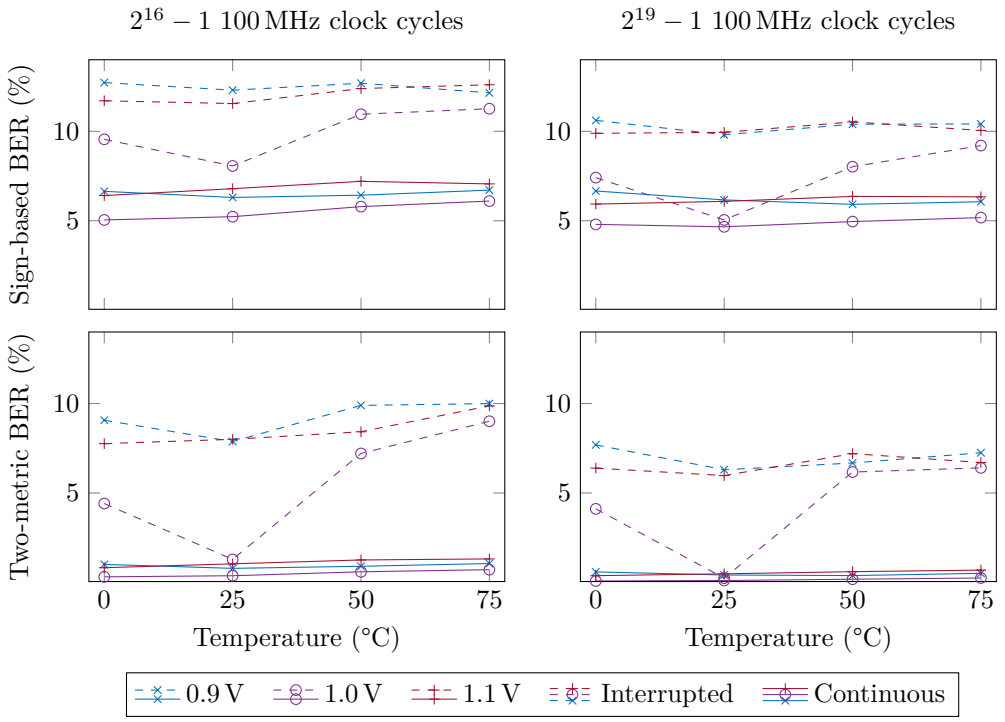


Figure 11: Evaluation of the BER for varying environmental temperatures and supply voltage with evaluation times of $2^{16} - 1$ (left) and $2^{19} - 1$ (right) clock cycles (100 MHz), for interrupted and continuous operating mode. The x-axis represents the temperature change, while different markers represent the evaluated supply voltages.

In general, a temperature change affects the propagation delay of circuits, thus changing the setup and hold time conditions. As these conditions shift, the probability of metastable states and quantization errors increases, leading to a higher BER.

It can also be seen that the relative increase of the BER is higher with the TMHD readout than with the sign-based one. We simulated ideal loops with increasing noise and found that while the TMHD approach is very effective at correcting errors, it does not provide much benefit beyond a certain error rate of about 10 %.

However, the absolute change in BER with TMHD readout is still lower than with sign-based readout. Furthermore, the measured increase in BER over temperature variation is lower or comparable to other architectures such as [MBC16, HWGH18].

Please note again that the aim of this work was not to design a PUF with more ideal characteristics compared to the SotA, but to modify an existing PUF in such a way that the resistance to SCA increases while the characteristics of the PUF remain almost the same. The minor variations by the added modification shown in Table 2 are evidence that the latter goal has been achieved, while the robustness against SCA will be analyzed next.

6 Side-Channel Attack Measurements

In Section 3.2 we have demonstrated the enhanced robustness against SCA of the proposed interrupted operation of RO-based PUFs by simulations. Next, a validation by means of measurements on the FPGA prototype is performed.

For the experiment, we assume an attacker with knowledge of the system design and the ability to repeatedly capture power traces. The underlying implementation is the

conventional continuously operating loop PUF as well as the proposed interruptible loop PUF. We assume that a fixed order of Hadamard challenges is applied during readout and that the attacker has no further control over the challenges applied. The same assumptions applied in the attacks on the original continuously running loop PUF, where it has been determined that temporal masking is insufficient for quantization schemes using the absolute value of the PUF response and a randomization of the challenge sequence would involve significant additional hardware [TKDP21].

The *measurement setup* to verify the side-channel security is based on a ChipWhisperer CW305 FPGA board which implements the continuously operating and the the interruptible loop PUFs. A LeCroy WaveSurfer 4104HD with a bandwidth of 1 GHz was connected directly to the pre-amplified power side-channel measurement output of the CW305 board to capture traces at 5 GSa/s. The loop evaluation time and therefore trace duration was set to 2.62 ms. During all measurements, only a single loop was active and the PRNG seed was kept constant. This was done to demonstrate the most powerful attacker, who is able to clearly observe the power draw of all challenges in sequence without any noise from other PUF instances.

The following experiments are then performed using this measurement setup:

- First, a naïve attack, directly using the spectrum of recovered traces, is analyzed in Section 6.1. While this attack is sufficient for the standard loop PUF, it will not reveal any information once the interrupt with sufficient frequency is activated.
- Second, Section 6.2 describes an improved attack that compensates for the influence of the interrupt on the oscillation. This allows a successful attack if the interrupt signal itself is known to the attacker.
- Section 6.3 lastly demonstrates how an attacker can be able to extract the interrupt signal via simple power analysis (SPA) for the second attack, and also a countermeasure against this attack. With this countermeasure in place, an extraction of the interrupt signal is no longer possible with high enough confidence, which prevents more advanced attacks, like the one in Section 6.2, from succeeding.

6.1 Naïve Side-Channel Analysis

When the interrupt signal is disabled, the implementation behaves like a regular loop PUF. We use this mode of operation to verify the results from [TKDP21], i.e., that without further protection measures, a spectrum-based attack on power traces is successful. For this experiment, we extend the loop PUF to a 64 bit implementation using otherwise the same design, which allows to attack more challenges per loop. The increase of delay elements in the loop reduces the nominal oscillation frequency to approx. 10 MHz; we thus also set $f_{\text{base}} = 10\,\text{MHz}$. This lower oscillation frequency is even more beneficial for a potential attack because it allows for a higher (relative) frequency resolution at the same oscilloscope sampling rate.

For the experiment, ten transient side-channel traces were acquired for each of the 128 challenges ($2 \cdot 64$ for challenge and bitwise inverse) in both modes (continuous/interrupted), which were then frequency-transformed and averaged. This includes the all-zero and the all-one challenge, even though they are not used in a bit derivation. Additional ten traces without active loop oscillations were used as a reference signal, whose average spectrum was subtracted from each of the 256 averaged challenge spectra to compensate for spurious peaks unrelated to the loop activity.

Results are shown in Fig. 12 for the continuous (top) and two interrupted (middle and bottom) operation modes of the one implemented loop PUF. Shown are – directly as measured and smoothed – the spectra of the power traces for the all-zero and the all-one challenges, which were chosen for their large frequency difference. In continuous

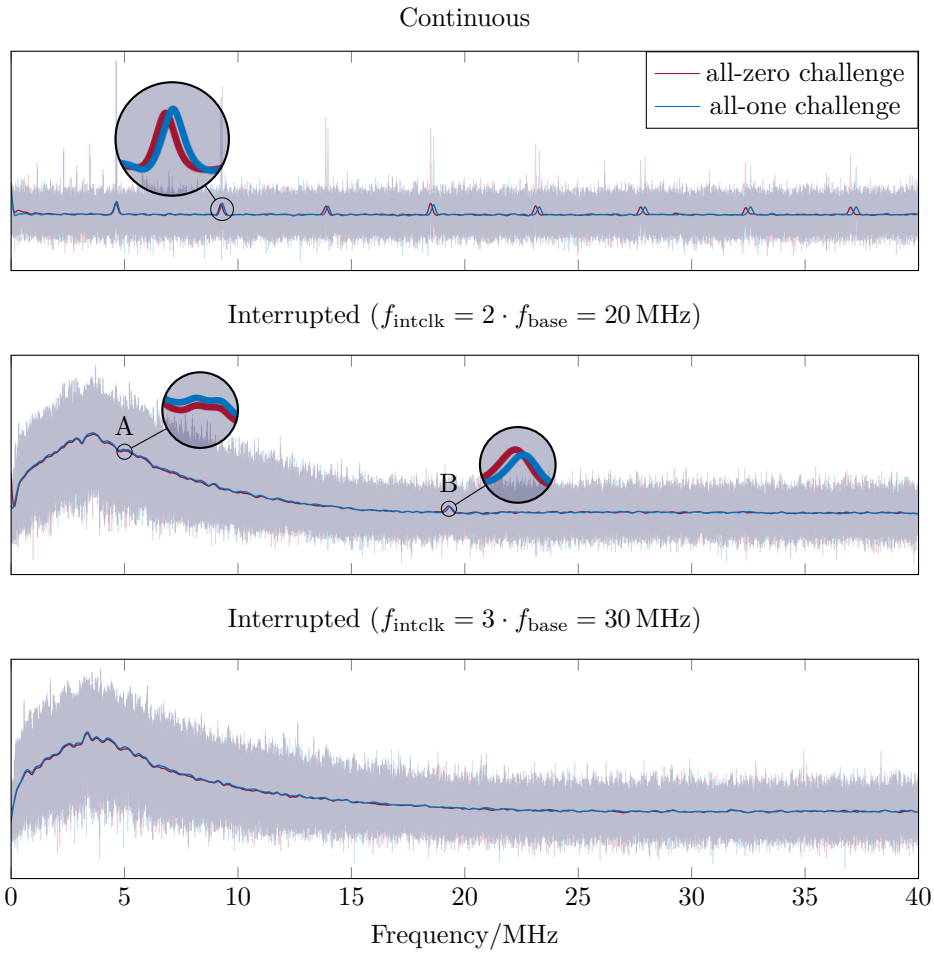


Figure 12: Averaged spectra for the all-zero and the all-one challenge of a single loop, which exhibit the largest oscillation frequency difference, as measured (in the background) and smoothed (colored traces). Without interrupt, the peaks and even their separation are clearly visible. With a 20 MHz interrupt, the peaks disappear except for one at B. At 30 MHz, no more peaks are visible.

mode, the frequency peaks are clearly visible; at higher harmonics, even the separation of both frequencies is directly visible confirming the vulnerability of the continuously operating loop PUF. By enabling the interrupted operation with a $2 \cdot f_{\text{base}} = 20\,\text{MHz}$ PRNG sequence, the spectrum largely changes, featuring a broad lobe at lower frequencies due to the randomness and distribution of the interrupt signal; the distinct frequency peaks disappear. However, two distinguishable features are indicated in the spectra:

A The spectra for the two exemplary challenges exhibit a slight difference in amplitude in the lower frequency range from 2...10 MHz. However, this amplitude difference was not found to contain information about the difference in oscillation frequency between challenges.

B After significant smoothing, a peak is visible at about 19.3 MHz, whose frequency shift correlates with the actual oscillation frequency. Over all challenges, this distinguishable peak allowed an attacker to recover quantized bits with an accuracy of about 70 % in our experiments.

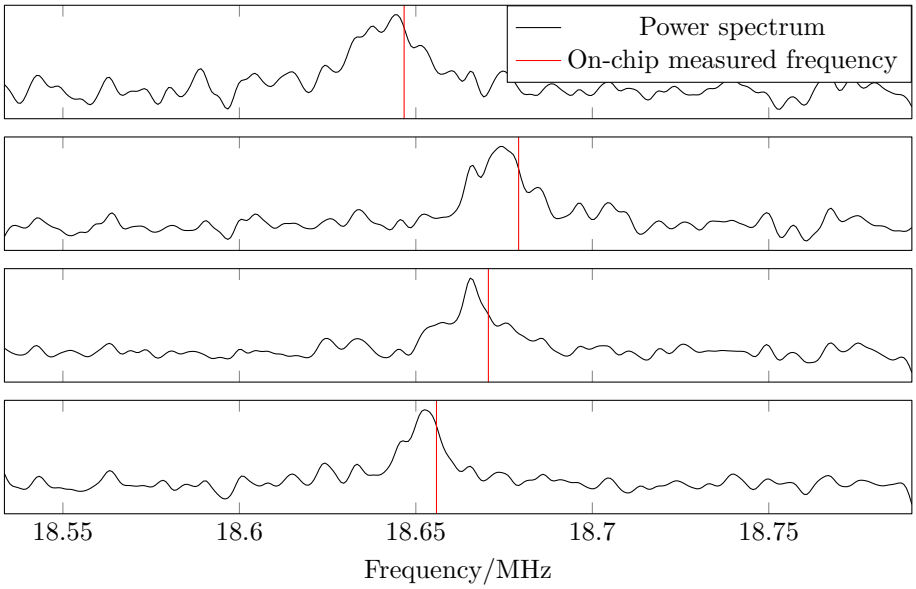


Figure 13: Spectra calculated for four exemplary challenges from signals reconstructed in the time domain using the known interrupt signal. 10 power traces were used for averaging. Peaks corresponding to the on-chip measured oscillator frequencies are visible.

The location of the peak (B) suggests it is a mixing product of the oscillation frequency (approx. 9.3 MHz) and the maximum output frequency (10 MHz) of the interrupt generator operated with a 20 MHz clock. A simple countermeasure was to increase the interrupt frequency, which is shown in the last plot of Fig. 12. Here, no obvious features could be revealed which were exploitable for an attack[1]. It is worth noting that peak for an interrupt frequency of 20 MHz was already visible by averaging two traces, while for an interrupt frequency of 30 MHz an average of 360 traces did not reveal any peak. In addition, the oscilloscope used oversamples the oscillation frequency by a factor of 100. Therefore, increasing these values is not expected to make the attack successful.

6.2 SCA With Knowledge of the Interrupt Signal

In the first naïve attack, it was assumed that the attacker had no knowledge of the interrupted operation of the loop PUF. Thus, they did not exploit any specific properties of its operation. In contrast, we might assume that the attacker has knowledge of the exact timing of the interrupt signal. Then, they can reconstruct a non-interrupted signal in the time domain by concatenating transient power trace segments based on the knowledge of the transient interrupt signal. Specifically, the attacker masks the parts of your side channel traces where the loop was off and combines the remaining samples where the loop was on into a new trace, which they then frequency-transform.

Fig. 13 shows resulting spectra from this attack on the interruptible loop PUF for four exemplary challenges. Despite the interrupted operation, the recombination of the power traces during on-times reconstructs the continuous operation. Consequently, the resulting spectra show clearly visible peaks corresponding to the oscillation frequencies as measured by the on-chip counter.

Given a known sequence of Hadamard challenges, the complete PUF response can be recovered by automatically identifying peaks in the spectra. Fig. 14 shows the attacker-

[1] A peak search near 24.3 MHz, the new mixing frequency, yielded an accuracy of 45…55 % and did thus not reveal information in a meaningful quantity.

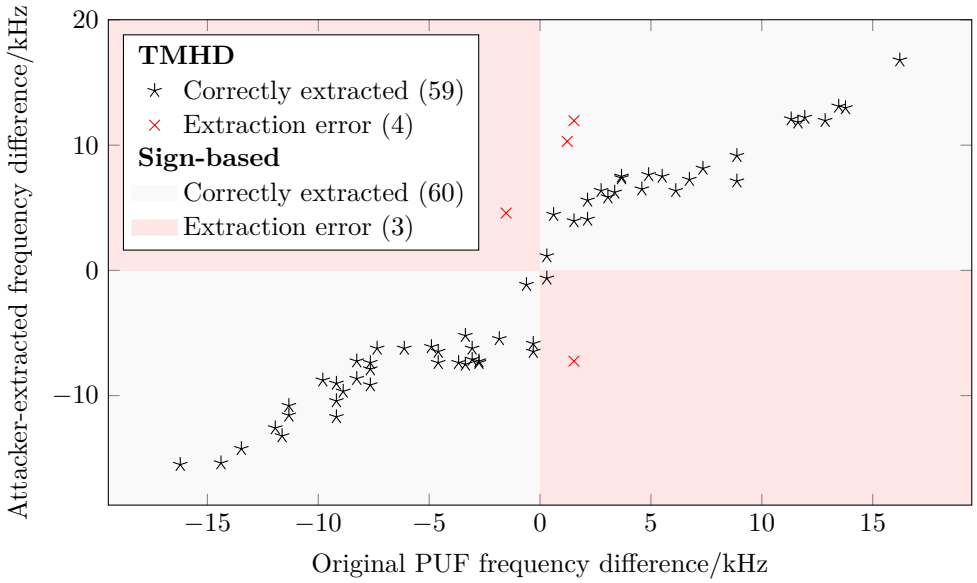


Figure 14: Frequency differences for all 63 challenge pairs of the single loop as determined by the PUF itself (x axis) and from the SCA measurements (y axis) with known interrupt signal. A very high correlation is apparent and consequently, the accuracy of recovered bits is over 90 % for both sign-based and TMHD quantization.

extracted frequency differences for all challenge pairs of one exemplary FPGA and one exemplary 64-bit loop PUF, where the bits are extracted with the sign-based and the TMHD method. The recovery accuracy of the attack is very good for both quantization schemes; not only the sign but also the amplitude of the frequency difference can thus be extracted with high accuracy. This can be seen Fig. 14 by comparing the x and y coordinates of the black crosses, which should be identical for a perfect reconstruction. On average, the measured and reconstructed frequency differences differ by 1.8 kHz. Further experiments showed that increasing the interrupt frequency does not prevent this attack; even at $20 \cdot f_{\text{base}} = 200\,\text{MHz}$ interrupt clock frequency, the attack was still successful.

Consequently, the proposed countermeasure against SCA by interrupting the operation of an oscillating loop PUF becomes ineffective once the attacker has knowledge of the exact interrupt signal. A deterministic interrupt signal, i.e., from a constant-seed PRNG, can thus not be used. A random signal solves this issue, however it still needs to be ensured that the interrupt signal itself does not leak. Thus, whether it is feasible for an attacker to get enough information about a random interrupt signal via SCA will be investigated next.

6.3 Techniques to Attack and Protect the Interrupt Signal

The interrupt signal has a substantial fan-out, driving each delay element individually, and thus exerts a considerable influence on power consumption. Therefore, it should be feasible to recover it using SPA. This possibility is demonstrated using a design comprising 72 16-bit loops, with a constant PRNG seed and, consequently, a deterministic interrupt signal.

Fig. 15a shows a time-domain power trace computed as the average of 320 interrupted loop PUF executions using the same interrupt signal, which is shown as a shaded background during the off-time of the oscillators. A dependency is visible, especially when the moving average is investigated: At each transition of the interrupt signal (blue to white or white to blue), the transition of the interrupt lines effects a large current spike, which can be

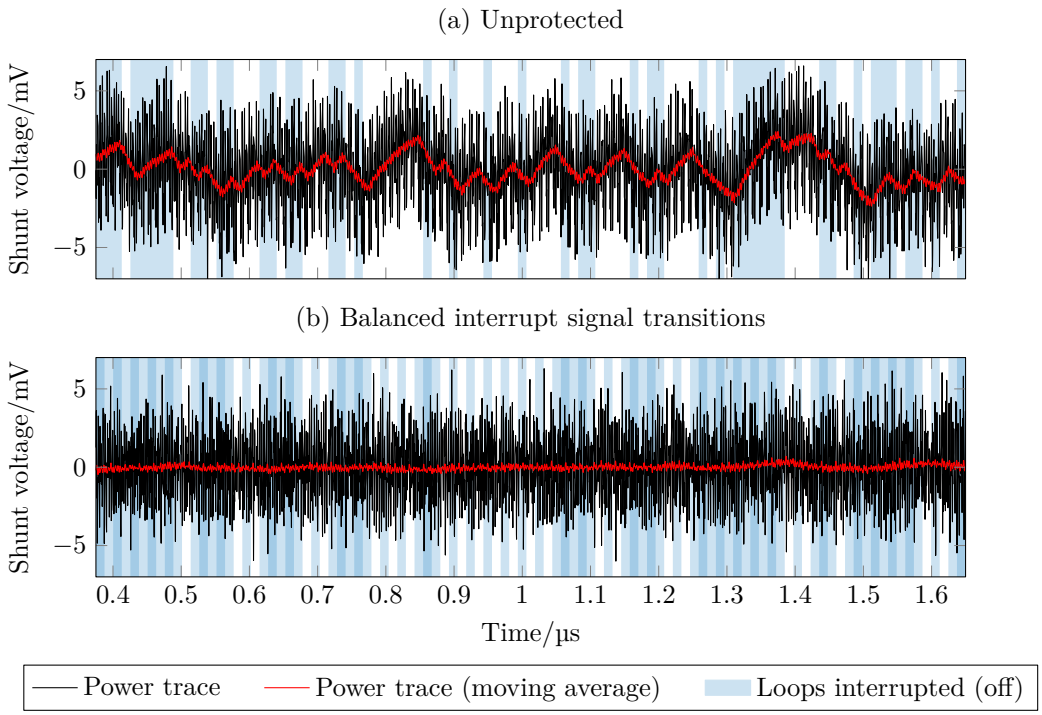


Figure 15: Average power traces for 320 measurements (10 repeated measurements for 32 challenges of a single 16 bit loop) with the same interrupt signal. With no further protection (a), the interrupt signal transitions can be directly extracted from a power traces using SPA. Ensuring balanced interrupt signal transitions (b) by using outputs 1 and 3 of Fig. 16 as interrupt lines effectively prohibits the attack.

seen as a voltage drop. During periods without interrupt signal transitions, the moving average shunt voltage rises again.

In order to combat the leakage of the interrupt signal, and in consequence the successful attack on the interrupted loop PUF, we suggest to follow a hiding strategy: In an array with multiple PUFs, which is expected to be needed in practical applications to generate a sufficient amount of secret bits, we generate different interrupt signals for two groups of loops with the intention to equalize their impact on the power traces. We aim for an implementation where the interrupt signal for one half of the loops is toggled in every cycle when the interrupt signal for the other half does not change. If the proportions of loops controlled by each interrupt are equal, exactly half of the interrupt lines toggle per clock cycle, balancing the power trace. The superimposed shaded areas in Fig. 15 show the states of the two interrupt signal variants. As there are now the same number of delay element interrupt inputs changing state per clock cycle, the moving average power trace no longer shows the interrupt signal.

The circuit shown in Fig. 16 can be used to generate different interrupt signals for parts of the loops from a common input signal. Outputs 1 and 3 are generated such that exactly one of them changes state each clock cycle. While we found the fan-out of the interrupt line itself to be by far the largest contributor to the leakage, there will also be differences in the power consumption of on- and off-state loops. To make it harder to exploit this fact, we use a hiding technique inspired by dual-rail logic (e.g. [DGBN09]): Outputs 2 and 4 of Fig. 16 are inverted copies of outputs 1 and 3, respectively. We split the two shares of PUFs again and use the four output signals of Fig. 16 each for a quarter of all loops. This

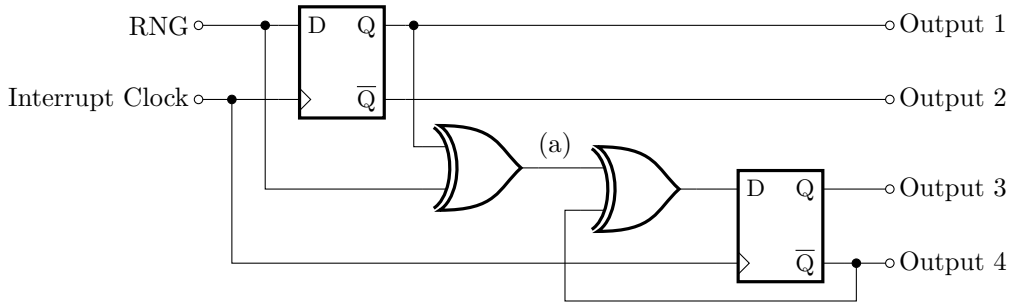


Figure 16: Proposed circuit for generating four distinct interrupt signals from a common random number generator input to protect against SPAs. The signal at (a) is low whenever output 1 does not change in a clock cycle and ensures that outputs 3 toggles in this case. Outputs 2 and 4 are inverted copies of outputs 1 and 3, respectively.

way, (i) in every clock cycle, half of the interrupt lines are toggled, and (ii) always half of the loops are off. As a consequence, the side channel leakage of the interrupt signal is minimized.

Note that—as typical for dual-rail and balancing strategies—this concept cannot completely eradicate side channel leakage, since small asymmetries in the implementation are unavoidable. However, practically using the remaining side channel leakage for an attack is complicated by two factors:

- The attack of Section 6.2 is fairly sensitive to errors in the interrupt signal: If the attacker only has an approximation of the interrupt sequence, bit errors accumulate as phase errors in the reconstructed time-domain signal. We found a bit error in the interrupt signal of 1 % to already seriously impact the success of the attack, which becomes infeasible at 2 %.
- The traces in Fig. 15 are averages from 320 power traces with the same interrupt signal. A practical implementation would only use the same interrupt signals for a challenge and its inverted copy, leaving an attacker with only two power traces to average.

Thus, with the countermeasure in place, it is infeasible to extract the interrupt signal with the required precision and the overall concept presented in this work constitutes a highly efficient countermeasure against SCA on oscillator based PUFs. Since only one loop was active in the experiments performed, this can be compared to a localized EM SCA, where the attacker can identify individual loops [MSSS11].

Still the attacker can bypass the interrupt signal protection (see Fig. 16) by a localized EM-SCA of the PRNG output directly. However, since the EM-SCA measurements are more noisy than power measurements [TDP20] and the attacker is limited to a maximum of two traces in the final architecture (challenge and its bitwise inverse), this will be quite challenging, especially since the reconstruction of the oscillation frequency is very sensitive to errors in the interrupt signal. Therefore, the presented countermeasure is expected to be effective against EM SCAs as well.

7 Applicability to Other PUF Architectures

The effectiveness of the presented countermeasure has been demonstrated using the loop PUF as an example. However, there are other oscillation-based PUFs, such as the RO or TERO PUF. The applicability of the countermeasure to them will be briefly discussed.

The RO PUF is very similar to the loop PUF with the difference that two ROs are evaluated in parallel with the same challenge applied. As a result, two frequency peaks will appear in the power spectra, revealing the magnitude information to the attacker. Interrupting both ROs would hide these peaks. Since the track-and-hold functionality is inside the ROs and the interrupt signal is buffered, mutual influence will not occur unless the ROs are correlated (which would corrupt the PUF anyway).

The TERO PUF evaluates the amount of oscillations until two coupled ROs settle to a stable value. In [TPI19] it was shown that an attacker is more interested in the duration of the oscillation than its frequency. Interrupting the oscillation, hides it from the attacker making it impossible to determine the duration. Quantization errors can speed up one of the signals, but it is very unlikely that the slower signal will always be preferred, which would distort the response.

Therefore, the countermeasure of an interruptible oscillator is also applicable to other oscillation-based PUF architectures. However, the length of the oscillator must be considered. Shorter oscillators are more likely to have quantization errors due to fewer track and hold elements. This can be compensated for by an increase in the evaluation time, which is shorter anyway. The maximum interrupt clock frequency must also be taken into account. This is limited by the available clock frequencies and the speed of the D-latches. Since the interrupt must be three times faster than the oscillators, this determines their minimum length.

8 Conclusion

This work presents the concept and implementation of an interrupting loop PUF architecture to protect the secret sign and magnitude information against power-trace SCA, by blurring the resulting PSD. Analysis of the implemented design showed that the architectural changes marginally affect the PUF quality. Performed SCAs were not able to recover the secret information without knowledge of the interrupt signal, while it was possible to correctly recover over 90 % of the bits when the interrupt signal is known. By adding interrupt signal protection, we have also hardened the proposed design against the extraction of the interrupt signal and found no remaining secret sign or magnitude information in the power-trace. Therefore, we conclude that the concept of a pseudo-random interruptible RO-based PUF successfully prevents power-trace SCA. It protects not only the sign-based quantization, but also hides the sensitive magnitude information used for multi-valued quantization or stabilization applications.

Acknowledgment

This work was supported in part by the German Federal Ministry of Education and Research through the Projects VE-VIDES under grant number 16ME0253 and VE-FIDES under grant number 16ME0257. The authors thank Robert Hesselbarth for his help with the large-scale PUF quality measurements.

References

[CBD+18] K-H Chuang, Erik Bury, Robin Degraeve, Ben Kaczer, Thomas Kallstenius, Guido Groeseneken, Dimitri Linten, and Ingrid Verbauwhede. A multi-bit/cell puf using analog breakdown positions in cmos. In *2018 IEEE International Reliability Physics Symposium (IRPS)*, pages P–CR. IEEE, 2018.

[CDGB12] Zouha Cherif, Jean-Luc Danger, Sylvain Guilley, and Lilian Bossuet. An Easy-to-Design PUF Based on a Single Oscillator: The Loop PUF. In *2012 15th Euromicro Conference on Digital System Design*, pages 156–162, Cesme, Izmir, Turkey, September 2012. IEEE.

[DGBN09] Jean-Luc Danger, Sylvain Guilley, Shivam Bhasin, and Maxime Nassar. Overview of dual rail with precharge logic styles to thwart implementation-level attacks on hardware cryptoprocessors. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*. IEEE, November 2009.

[DGS19] Jean-Luc Danger, Sylvain Guilley, and Alexander Schaub. Two-metric helper data for highly robust and secure delay pufs. In *2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI)*, pages 184–188. IEEE, 2019.

[DHWG20] Ding Deng, Shen Hou, Zhenyu Wang, and Yang Guo. Configurable ring oscillator puf using hybrid logic gates. *IEEE Access*, 8:161427–161437, 2020.

[Fis24] Robert FH Fischer. Helper data schemes for coded modulation and shaping in physical unclonable functions. *arXiv preprint arXiv:2402.18980*, 2024.

[GCVDD02] Blaise Gassend, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 148–160, 2002.

[GTS15] Fatemeh Ganji, Shahin Tajik, and Jean-Pierre Seifert. Why attackers win: on the learnability of xor arbiter pufs. In *Trust and Trustworthy Computing: 8th International Conference, TRUST 2015, Heraklion, Greece, August 24-26, 2015, Proceedings 8*, pages 22–39. Springer, 2015.

[HBF08] Daniel E Holcomb, Wayne P Burleson, and Kevin Fu. Power-up sram state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, 2008.

[HMBO19] Andreas Herkle, Holger Mandry, Joachim Becker, and Maurits Ortmanns. In-depth analysis and enhancements of ro-pufs with a partial reconfiguration framework on xilinx zynq-7000 soc fpgas. In *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 238–247. IEEE, 2019.

[HWGH18] Robert Hesselbarth, Florian Wilde, Chongyan Gu, and Neil Hanley. Large scale RO PUF analysis over slice type, evaluation time and temperature on 28nm Xilinx FPGAs. In *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 126–133, Washington, DC, April 2018. IEEE.

[IHL+19] Vincent Immler, Matthias Hiller, Qinzhi Liu, Andreas Lenz, and Antonia Wachter-Zeh. Variable-length bit mapping and error-correcting codes for higher-order alphabet pufs—extended version. *Journal of Hardware and Systems Security*, 3:78–93, 2019.

[MBC16] Cédric Marchand, Lilian Bossuet, and Abdelkarim Cherkaoui. Design and characterization of the tero-puf on sram fpgas. In *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 134–139. IEEE, 2016.

[MCMS10] Abhranil Maiti, Jeff Casarona, Luke McHale, and Patrick Schaumont. A large scale characterization of ro-puf. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 94–99. IEEE, 2010.

[MHM+20] Holger Mandry, Andreas Herkle, Sven Müelich, Joachim Becker, Robert FH Fischer, and Maurits Ortmanns. Normalization and multi-valued symbol extraction from ro-pufs for enhanced uniform probability distributions. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(12):3372–3376, 2020.

[MSSS11] Dominik Merli, Dieter Schuster, Frederic Stumpf, and Georg Sigl. Semi-invasive em attack on fpga ro pufs and countermeasures. In *Proceedings of the Workshop on Embedded Systems Security*, pages 1–9, 2011.

[MV10] Roel Maes and Ingrid Verbauwhede. Physically unclonable functions: A study on the state of the art and future research directions. *Towards Hardware-Intrinsic Security: Foundations and Practice*, pages 3–37, 2010.

[NSJ+18] Phuong Ha Nguyen, Durga Prasad Sahoo, Chenglu Jin, Kaleel Mahmood, Ulrich Rührmair, and Marten Van Dijk. The interpose puf: Secure puf design against state-of-the-art machine learning attacks. *Cryptology ePrint Archive*, 2018.

[PKS22] Jaehan Park, ByungJun Kim, and Jae-Yoon Sim. A ber-suppressed puf with an amplification of process mismatch effect in an oscillator collapse topology. *IEEE Journal of Solid-State Circuits*, 57(7):2208–2219, 2022.

[SD07] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual design automation conference*, pages 9–14, 2007.

[SP24] Niklas Stein and Michael Pehl. Leakage sources of the icloopuf: Analysis of a side-channel protected oscillator-based puf. In Romain Wacquez and Naofumi Homma, editors, *Constructive Side-Channel Analysis and Secure Design*, pages 157–176, Cham, 2024. Springer Nature Switzerland.

[TDF+17] Shahin Tajik, Enrico Dietz, Sven Frohmann, Helmar Dittrich, Dmitry Nedospasov, Clemens Helfmeier, Jean-Pierre Seifert, Christian Boit, and Heinz-Wilhelm Hübers. Photonic side-channel analysis of arbiter pufs. *Journal of Cryptology*, 30:550–571, 2017.

[TDP20] Lars Tebelmann, Jean-Luc Danger, and Michael Pehl. Self-secured puf: Protecting the loop puf by masking. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 293–314. Springer, 2020.

[TDP22] Lars Tebelmann, Jean-Luc Danger, and Michael Pehl. Interleaved challenge loop puf: A highly side-channel protected oscillator-based puf. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 69(12):5121–5134, December 2022.

[TKDP21] Lars Tebelmann, Ulrich Kühne, Jean-Luc Danger, and Michael Pehl. Analysis and protection of the two-metric helper data scheme. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 279–302. Springer, 2021.

[TNH+14] Shahin Tajik, Dmitry Nedospasov, Clemens Helfmeier, Jean-Pierre Seifert, and Christian Boit. Emission analysis of hardware implementations. In *2014 17th Euromicro Conference on Digital System Design*, pages 528–534. IEEE, 2014.

[TPI19] Lars Tebelmann, Michael Pehl, and Vincent Immler. Side-channel analysis of the tero puf. In *Constructive Side-Channel Analysis and Secure Design: 10th International Workshop, COSADE 2019, Darmstadt, Germany, April 3–5, 2019, Proceedings 10*, pages 43–60. Springer, 2019.

[Xil13] Xilinx. *Xilinx 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide for HDL Designs (UG768)*, 2013.

[ZHW+21] Yin Zhang, Zhangqing He, Meilin Wan, Jiuyang Liu, Haoshuang Gu, and Xuecheng Zou. A sc puf standard cell used for key generation and anti-invasive-attack protection. *IEEE Transactions on Information Forensics and Security*, 16:3958–3973, 2021.

A Mathematical Description of the Countermeasure

A mathematical description of the random interruptible oscillator is quite complex. Since the oscillation is not only stopped, but the current value is held, a simple multiplication with the interrupt is not sufficient to model the behavior. Several things must be considered: First, stopping and holding the current value randomly prolongs the duration of each period of the oscillation. Second, the ratio of on to off states changes from 50 : 50 to an unpredictable random ratio, because the interrupt sequence during the on state is different from the sequence during the off state. Third, to model stopping and resuming, the individual periods of the oscillator signal must be shifted by the duration of previous interrupts (see the yellow-marked signal part in Fig. 3c).

For simplicity, only the on states of the oscillation are considered, while the prolongation of the off states can be seen as a shift of the following period. Thus the interrupted oscillation $\mathrm{RO}(t)_{\mathrm{int}}$ can be seen as sum of randomly shifted *rect* signals of random width:

$$\mathrm{RO}(t)_{\mathrm{int}} = \sum_{i=0}^{\infty} \mathrm{rect}\left(\frac{t - (i \cdot T_{ro} + T_{PS}(i))}{T_{PW}(i)}\right). \tag{4}$$

The random shift $T_{PS}(i)$ and width $T_{PW}(i)$ both depend on the output of the PRNG generating the interrupt and the interrupt clock frequency $f_{\mathrm{clk}} = 1/T_{\mathrm{clk}}$. The duration of the original oscillation of the frequency $f_{\mathrm{ro}} = 1/T_{\mathrm{ro}}$ cannot be shortened, but only prolonged. Since the output of the PRNG changes with the frequency f_{clk}, the width $T_{PW}(i)$ of the current period, i.e. the rect signal, expands by a random multiple of T_{clk}:

$$T_{PW}(i) = \frac{T_{\mathrm{ro}}}{2} + P_w(i) \cdot T_{\mathrm{clk}} \tag{5}$$

with $P_w(i)$ being the amount of generated ones of the PRNG output during the current rect signal. The product of $P_w(i) \cdot T_{clk}$ is the time the interrupt is high. The shift of each rect signal in (4) depends on two factors: First, the basic shift in the continuous signal $(i \cdot T_{ro})$ and second the random interrupts:

$$T_{PS}(i) = P_s(i) \cdot T_{\mathrm{clk}}, \tag{6}$$

with $P_s(i)$ being the amount of generated ones of the PRNG output from the start of the evaluation till the start of the current rect.

The PSD used by the attacker can be calculated from the Fourier-transformed autocorrelation function. Thus the PSD of $\mathrm{RO}(t)_{int}$ corresponds to

$$\begin{aligned} \mathrm{PSD}(f) &= \sum_{i=0}^{\infty} |T_{PW}(i) \cdot si^2(T_{PW}(i) \cdot f) \cdot e^{-j2\pi f 2(i \cdot T_{ro} + T_{PS}(i))}| \\ &= \sum_{i=0}^{\infty} T_{PW}(i) \cdot si^2\left(T_{PW}(i) \cdot f\right) \end{aligned} \tag{7}$$

Thus, the PSD of the interrupted oscillator is the sum of si^2 functions of random width, depending on the PRNG sequence. As long as two interrupt sequences generated by one or multiple PRNGs are uncorrelated, the individual widths are also uncorrelated. The PSD adds up to white noise, which hides the oscillator frequency.

This model was found to be in good agreement with the measured results.