A Deep Analysis of two Glitch-Free Hardware Masking Schemes SESYM and LMDPL

Nicolai Müller[1], Daniel Lammers[1] and Amir Moradi[2]

[1] Ruhr University Bochum, Horst Görtz Institute for IT Security, Bochum, Germany
firstname.lastname@rub.de
[2] Technische Universität Darmstadt, Darmstadt, Germany
firstname.lastname@tu-darmstadt.de

Abstract. In the context of masking, which is the dominant technique for protecting cryptographic hardware designs against Side-Channel Analysis (SCA) attacks, the focus has long been on the design of masking schemes that guarantee provable security in the presence of glitches. Unfortunately, achieving this comes at the cost of increased latency, since registers are required to stop glitch propagation. Previous work has attempted to reduce latency by eliminating registers, but the exponential increase in area makes such approaches impractical. Some relatively new attempts have used Dual-Rail Pre-charge (DRP) logic styles to avoid glitches in algorithmically masked circuits. Promising approaches in this area include LUT-based Masked Dual-Rail with Pre-charge Logic (LMDPL) and Self-Synchronized Masking (SESYM), presented at CHES 2020 and CHES 2022 respectively. Both schemes allow masking of arbitrary functions with only one cycle latency. However, even if glitches no longer occur, there are other physical defaults that may violate the security of a glitch-free masked circuit. The imbalanced delay of dual rails is a known security problem for DRP logic styles such as Wave Dynamic Differential Logic (WDDL), but is not covered by the known security models, e.g., robust probing model.

In this work, we illustrate that imbalanced signal delays pose a threat to the security of algorithmically masked circuits implemented with DRP logic, both in theory and practice. Notably, we underscore the security of LMDPL even when delays are taken into account, contrasting with the vulnerability observed in SESYM under similar conditions. Consequently, our findings highlight the critical importance of addressing imbalanced delays in the design of masked circuits using DRP logic. In particular, our findings motivate the need for an appropriate security model, and imply that relying solely on the probing security model and avoiding glitches may be insufficient to construct secure circuits.

Keywords: Side-Channel Analysis · Masking · Hardware · Dual-Rail Pre-charge Logic · Robust Probing Model

1 Introduction

Although first discovered by Kocher [Koc96] over 25 years ago, Side-Channel Analysis (SCA) attacks still pose a significant threat to the confidentiality of sensitive data on cryptographic devices. Following Kocher's discovery, several attacks have been developed over the years that take advantage of various physical properties, also known as side channels [KJJ99, GMO01, GST14, HS13, GSST07]. As a result, both industry and academia have devoted considerable effort to developing effective countermeasures to prevent SCA attacks. In particular, masking [CJRR99], an algorithmic approach inspired from secret sharing [Sha79], stands out.

Masking randomizes all sensitive intermediates of an algorithm by splitting them into multiple shares, such that no incomplete set of shares reveals information about any sensitive intermediate. A major advantage of masking over other countermeasures is that formal security models abstract its physical security, simplifying the realization and verification of a masked implementation and accelerating the design process. The basic d-probing model defines an abstract adversary capable of placing probes on up to d arbitrary circuit wires [ISW03]. Each probe records a stable signal at a fixed point in time. In addition, a masked circuit is considered secure under the d-probing model if the adversary is unable to obtain any information about sensitive variables.

However, a masked cryptographic design that is recognized as secure under the d-probing model may become vulnerable when implemented in practice. For example, Mangard et al. [MPO05] mounted successful first-order Differential Power Analysis (DPA) attacks on theoretically secure masked Advanced Encryption Standard (AES) designs [AG01, OMPR05] implemented on an Application-Specific Integrated Circuit (ASIC) prototype. It has been conjectured that unexpected combinational recombinations due to different delays in non-ideal circuits, so-called glitches, leak information about sensitive variables. In the underlying circuits, glitches propagate through the S-box computation, resulting in a toggle count that depends on (unmasked) sensitive variables [OMPR05]. The d-probing model itself assumes an ideal circuit and does not cover physical defaults, e.g., glitches. Hence, a d-probing secure design may still leak information due to glitches. To verify the probing security even in the presence of glitches, the robust d-probing model [FGP+18] was developed extending the traditional d-probing model. Among other parameters, the robust d-probing model technically introduces glitch-extended probes, which allow an adversary to gain access to all intermediate values that could potentially be revealed when probing an arbitrary wire.

The robust probing model covers glitches by assuming a worst-case scenario, taking into account every possible occurrence of a glitch. However, only certain glitches occur in a physical circuit, while others never do or do not leak sensitive information. Consequently, worst-case models can be overly conservative because they account for physical defaults that may not occur in an actual implementation of a design. Nevertheless, worst-case models are necessary to prove the security of arbitrary circuits without fine-grained characterization. In particular, the proofs based on models are independent of how the circuit is realized, e.g., irrespective of the placement of logical cells and the routing of their interconnections.

One line of research has focused on designing glitch-immune circuits, i.e., circuits that remain probing secure for every possible occurrence of a glitch. However, for the purpose of power equalization, i.e., hiding [MOP07], other works have previously attempted to design glitch-free circuits, where glitches are entirely prevented. A well-studied approach in this area is the application of Dual-Rail Pre-charge (DRP) logic, where a signal x is transmitted by its original wire, denoted as x_{t}, and another wire, denoted as x_{f}, carrying the complementary signal. After initializing a zero state $(x_{\mathrm{t}}, x_{\mathrm{f}}) = (0, 0)$, switching from this state to a valid state, which is either $(x_{\mathrm{t}}, x_{\mathrm{f}}) = (1, 0)$ or $(x_{\mathrm{t}}, x_{\mathrm{f}}) = (0, 1)$, requires a constant amount of energy that is independent of the data. To enforce the zero state, all signals are pre-charged, i.e., set to zero, before the inputs switch to a valid state. Moreover, DRP allows the construction of glitch-free gadgets, which are the basis of Wave Dynamic Differential Logic (WDDL) [TV04a], a DRP-based power-equalization technique that relies on custom gates built by standard cell libraries. While circuits, which are solely based on WDDL gadgets, are glitch-free, their dynamic power consumption still depends on the processed data [SS06]. This is because the time at which a WDDL gadget produces its valid output depends on the given inputs, leading to a data-dependent time-of-evaluation. As a result, the power consumption of a WDDL circuit still exhibits some data dependency.

Although WDDL is not sufficient as a stand-alone SCA countermeasure, its glitch-free nature – achieved through the use of DRP logic – offers promising possibilities in combination with other countermeasures, such as masking. Recent publications at CHES 2020 [SBHM20] and CHES 2022 [NGPM22] suggest removing the register stages whose sole purpose is to stop glitch propagation when the circuit is glitch-free. Thus, a combination of DRP and masking should allow the construction of masking schemes which implement arbitrary functions within a single clock cycle. Such constant-time, low-latency schemes are of particular interest – especially to industry – because of their high potential for real-time applications, such as fast memory encryption. Moreover, if the approaches are viable, proving the security of a masked implementation is potentially simplified, since the underlying security model does not need to consider glitches. In addition to stopping the propagation of glitches, however, registers have another function. They synchronize intermediate states *independent of data* as they are controlled by a global synchronization signal, e.g., a clock signal. Without this synchronization, time differences can propagate through the circuit, ultimately leading to a data-dependent time-of-evaluation. We remark that the robust probing model does not cover timing differences in DRP circuits. Currently, these effects are either ignored [NGPM22] or examined in an informal and argumentative manner [SBHM20].

Our Contributions. This work addresses the impact of early propagation effects and routing imbalances on the physical security of algorithmically masked glitch-free circuits. Here, we should highlight that we do not deal with "masked dual-rail pre-charge logic styles" MDPL and iMDPL [PM05, PKZM07], where a single mask bit is shared between all gates of the entire circuit. Their security has already been criticised in [SS06, MKEP12]. We indeed emphasize the inconsistency of the (robust) probing model with reality when evaluating glitch-free circuits that implement algorithmically masked designs. Our contributions can be summarized as follows.

- We re-assess the security analysis of two practical masking schemes that use DRP logic and are designed to be glitch-free: LUT-based Masked Dual-Rail with Pre-charge Logic (LMDPL) [SBHM20] and Self-Synchronized Masking (SESYM) [NGPM22]. Alongside glitches, this re-evaluation incorporates the effect of imbalanced delays which cover data-dependent time-of-evaluation as well. In short, our findings imply the security of LMDPL, while this is not the case for SESYM, mainly due to routing imbalances which is hard to avoid.

- We validate and quantify our theoretical observations through experimental measurements on Field Programmable Gate Array (FPGA) prototypes. The results confirm that significant leakage is exhibited by the SESYM designs constructed by the original authors. Furthermore, the observed leakage can be readily detected without the need for artificially enhancing the susceptibility of most of the hardware designs. This underscores that the observed vulnerabilities are not solely of academic interest, but can indeed compromise the security of real-world designs.

2 Background

2.1 Notations

We denote single-bit variables by lower case letters, e.g., $x \in \mathbb{F}_2$, vectors by upper case letters, e.g., $X \in \mathbb{F}_2^n$, and sets by a calligraphic font like $\mathcal{S}$. For masked (shared) variables, we use superscripts to denote a particular share domain, e.g., x^0 as the first share of x. If a variable is represented by two rails, we use subscripts to denote specific rails. In particular, x_t denotes the original and x_f the corresponding inverted rail, both presenting $x : (x_\mathrm{t}, x_\mathrm{f})$. We use $\oplus$ to denote the addition in $\mathbb{F}_2$.

2.2 Dual-Rail Pre-charge (DRP) Logic

The goal of DRP logic is to equalize the power consumption of a circuit by guaranteeing a constant number of toggles per clock cycle. Therefore, a DRP circuit is expected to hide the data-dependent power consumption below the noise level in the amplitude domain, making DPA attacks harder to succeed. Such a circuit processes each signal by means of two wires. The first wire drives the original value and the second one its complement. This encoding, shown below, guarantees that one rail always carries logical one, while the other rail carries logical zero.

$$\left.\begin{matrix} x_{\mathrm{t}} = 0 \\ x_{\mathrm{f}} = 0 \end{matrix}\right\} x = \textsc{null}, \quad \left.\begin{matrix} x_{\mathrm{t}} = 0 \\ x_{\mathrm{f}} = 1 \end{matrix}\right\} x = 0, \quad \left.\begin{matrix} x_{\mathrm{t}} = 1 \\ x_{\mathrm{f}} = 0 \end{matrix}\right\} x = 1, \quad \left.\begin{matrix} x_{\mathrm{t}} = 1 \\ x_{\mathrm{f}} = 1 \end{matrix}\right\} x = \textsc{invalid}$$

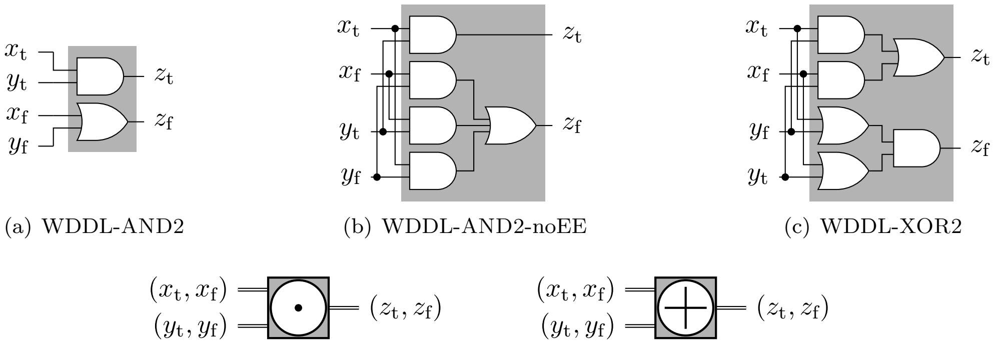
To balance the power consumption, it is necessary to apply a pre-charge technique that sets all circuit wires, i.e., both rails of all signals, to zero, i.e., to the NULL state. This so-called pre-charge phase alternates with an evaluation phase in which the actual outputs of the circuit are computed. Since all signals switch from NULL to a valid state, and all valid states result in a single toggle per signal, the number of toggles in the entire circuit (and ideally the power consumption) becomes constant for every possible input value.

2.3 Wave Dynamic Differential Logic (WDDL)

WDDL [TV04a] is a well-known design technique that replaces the standard cells of the given circuit with gadgets that are also composed of standard cells. The goal is to design a circuit with constant power consumption regardless of the data being processed. The implementations of different WDDL gadgets are depicted in Figure 1. It has been shown by Suzuki et al. in [SS06] that WDDL gadgets may fail to keep the instantaneous power consumption constant when the inputs arrive at different time instances. The underlying concept is referred to as the *early propagation* effect or the *data-dependent* time-of-evaluation.

Definition 1 (Early Propagation Effect)**.** Some DRP gadgets can uniquely determine their output based on a subset of their inputs, without waiting for all the inputs to arrive. Accordingly, such gadgets can propagate their output early.

Example 1 (Early Propagation Effect)**.** Let us consider the OR gate in the WDDL-AND2 gadget shown in Figure 1(a) with the input signals x_{f} and y_{f} and output signal z_{f}. If we assume that $x_{\mathrm{f}} = 1$, then $z_{\mathrm{f}} = 1$ is propagated without any knowledge about y_{f}.


x_{t}
y_{t}
x_{f}
y_{f}
z_{t}
z_{f}
$(x_{\mathrm{t}}, x_{\mathrm{f}})$
$(y_{\mathrm{t}}, y_{\mathrm{f}})$
$(z_{\mathrm{t}}, z_{\mathrm{f}})$

(a) WDDL-AND2

(b) WDDL-AND2-noEE

(c) WDDL-XOR2

(d) WDDL-AND2 symbol

(e) WDDL-XOR2 symbol

Figure 1: WDDL gadgets.

To avoid early propagation, Bahsin et al. [BGF+10] introduced WDDL-noEE, shown in Figure 1(b). However, this only prevents early propagation when the circuit traverses from the pre-charge to the evaluation phase, whereas a global pre-charge signal (connected to all gadgets) can mitigate the early propagation from the evaluation to the pre-charge phase. Subsequently, Moradi et al. [MI14] introduced AWDDL, an asynchronous variant of WDDL that avoids the early propagation effect in both phases, assuming no delay imbalances. In addition, the authors presented a custom router that minimizes – but does not completely eliminate – the risk of routing imbalances by taking care of placement in an FPGA design. It is worth noting that the fat-wire approach [TV04b] also aims to avoid delay imbalances in ASIC design flows, although the capacitances of dual-rail wires in a fabricated circuit may still differ slightly.

2.4 Masking

Masking, based on the concept of secret sharing [Sha79], is the predominant approach to protect hardware circuits against SCA attacks [CJRR99]. In *d-th order Boolean masking*, every sensitive variable $X \in \mathbb{F}_2^n$ is randomized by splitting it into a set of $d+1$ *shares* $\{X^0, ..., X^d\} \in (\mathbb{F}_2^n)^{d+1}$, each of which follows a uniform distribution. Exemplary, every sharing can be initialized by uniformly sampling the first d shares at random and computing the last one as $X^d = \left(\bigoplus_{i=0}^{d-1} X^i\right) \oplus X$. To achieve the desired security with $d+1$ shares, the target algorithm is transformed into a masked algorithm by performing all operations on the shared representation of sensitive variables.

Domain-Oriented Masking (DOM). The underlying technique of DOM allows the creation of d-th order secure Boolean masked circuits by utilizing exactly $d+1$ shares [GMK16]. The idea is to split a circuit into $d+1$ independent share domains, receiving at most one share per variable. While linear operations achieve domain independence by processing each share individually, non-linear operations, such as the first-order secure DOM multiplier shown in Figure 4(a), must combine shares across domain boundaries. To maintain domain independence, some intermediates are blinded by fresh randomness and synchronized by registers to avoid combining both shares of the same variable.

2.5 Security Models

The basic d-probing model [ISW03] defines an adversary who gains access to up to d intermediates of an ideal circuit. Based on the capabilities of the so-called *d-probing adversary*, *d-probing security* is defined as follows.

Definition 2 (d-Probing Security)**.** A circuit is d-probing secure iff every d-tuple of intermediates gives no information about any secret variable.

To model the observation set, an attacker places up to d probes on arbitrary wires. Each probe then records a stable signal at a given time instance (reflected by the meaning of an ideal circuit). The robust d-probing model [FGP+18] builds on top of the above-given d-probing model extending it to cover physical defaults, i.e., glitches, transitions, and couplings.

Definition 3 (Robust d-Probing Security [FGP+18])**.** A circuit is robust d-probing secure iff every d-probing set, containing up to d probes extended by glitches, transitions and couplings, gives no information about any secret variable.

In this work, we consider the robust probing model including only glitches, i.e., the glitch-extended d-probing model. Note that in DRP circuits, transitions are avoided by alternating pre-charge/evaluation phases while couplings [CBG+17] cannot be evaluated at the gate level. Hence, below we only deal with formal definitions related only to glitches.

Definition 4 (Glitch)**.** If inputs of a logic gate arrive asynchronously, the gate processes an unstable input state, which may result in a transient change in the output. Hence, the gate may propagate an unexpected output before reaching its stable state. We refer to such a transient faulty output as a glitch [MPG05].

Definition 5 (Glitch-extended Probing Model)**.** For any circuit, we can model the effect of glitches using glitch-extended probes, so that a glitch-extended probe on wire w allows the adversary to observe all stable signals contributing to w. This means that a probe placed on w propagates backwards through the combinational circuit constructing w until reaching the circuit's primary inputs or register outputs. These stable signals make a probing set as the result of extending w via glitches.

2.6 Composability

The design and evaluation of provably robust probing secure circuits becomes infeasible with increasing design complexity, e.g., for higher-order security, since composing standalone secure modules does not necessarily lead to a secure design [CPRR13]. To overcome this difficulty, tiny and provably secure building blocks have been introduced, which can be composed to realize provably secure functions of arbitrary orders. We refer to these building blocks as *composable gadgets* [CGLS21]. Such gadgets allow for the transformation of any unprotected gate-level netlist into a masked one by simply replacing any unprotected gate with a composable gadget. In practice, the transformation can be done automatically, e.g., by AGEMA [KMMS22].

Definition 6 (Composability)**.** A set of gadgets, which are individually secure under a given security model, are composable iff all their possible combinations are still secure under the same security model.

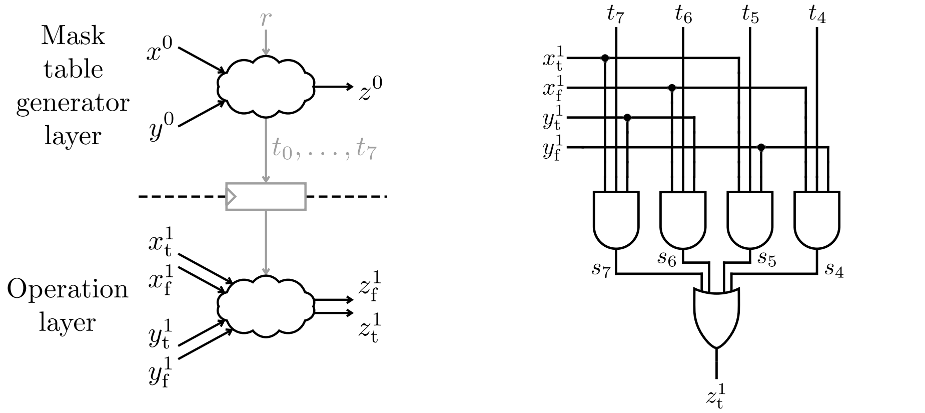
Therefore, an arbitrary circuit made out of gadgets is robust d-probing secure if all underlying gadgets are composable under the robust d-probing model.

3 Single-Cycle Masking in Hardware

In addition to area and randomness requirements, efficiency with respect to latency tends to be relevant for conventional masking schemes, especially those based on composable gadgets [CGLS21]. We review two single-cycle hardware masking schemes LMDPL and SESYM, both of which incorporate DRP principles.

3.1 LMDPL

LMDPL [LMW14] enables first-order secure masking of arbitrary circuits with a constant latency, i.e., one cycle for pre-charge and one cycle for evaluation, which can be packed into a single clock cycle. In a follow-up work, Sasdrich et al. [SBHM20] provided some proofs for the security of LMDPL under the glitch-extended probing model and claimed that gadgets based on LMDPL should be composable. Each gadget processes two shares per variable, and a single fresh mask blinds one share if the gadget is non-linear. All LMDPL gadgets are divided into two layers, as depicted in Figure 2(a). The mask table generation layer of non-linear gadgets receives the first share of all input variables (x^0, y^0) and a fresh mask r to compute some intermediate values $t_0, \ldots, t_7$ while the first output share is derived solely from the fresh mask, i.e., $z^0 = r$. The operation layer takes the second shares in their dual-rail representation $(x^1_\mathrm{t}, x^1_\mathrm{f}, y^1_\mathrm{t}, y^1_\mathrm{f})$ and computes z^1_t and z^1_f. Since the operation layer is built using DRP logic, it is guaranteed that no glitches occurs.



(a) LMDPL gadget layers, gray colored components are needed if the gadget is non-linear.

(b) One output rail of a non-linear LMDPL gadget operation layer. z^1_f is generated analogously.

Figure 2: An overview of LMDPL gadgets [SBHM20].

In case of a non-linear gadget realizing function $F(.,.)$ over two variables (e.g., 2-input AND gate), a register stage separates both layers, and the fresh mask blinds all masked table outputs $t_0, \ldots, t_7$ which are computed with $i, j \in \{0, 1\}$ as follows.

$$t_{4+2i+j} = F(x^0 \oplus j, y^0 \oplus i) \oplus r, \qquad t_{2i+j} = t_{4+2i+j} \oplus 1$$

The operation layer processes all $t_0, \ldots, t_7$ by instantiating two 4-to-1 multiplexers while x^1 and y^1 act as a 2-bit select signal (cf. Figure 2(b)). We remark that the authors analyzed the timing of the multiplexers and claim that their design satisfies data-independent time-of-evaluation. In addition to the second share variables $(x^1_\mathrm{t}, x^1_\mathrm{f}, y^1_\mathrm{t}, y^1_\mathrm{f})$, all $t_0, \ldots, t_7$, which are stored in registers, should follow a pre-charge/evaluation fashion per clock cycle to guarantee the absence of glitches.

3.2 SESYM

Nagpal et al. [NGPM22] described a generic procedure for converting any Boolean-masked circuit into a single-cycle glitch-extended d-probing secure circuit without the need for additional randomness. The process involves converting single-rail inputs to dual-rail and employing DRP gates to achieve single-cycle processing. A so-called completion detector module (a product-of-sum form of all dual-rail outputs) detects the completion of the evaluation phase for all DRP output variables, and immediately induces a pre-charge wave into the masked circuit. In parallel, an array of Muller C-elements maintains the stable output signals and acts as a dual-to-single-rail converter. Figure 3 gives a rough overview of the scheme.

The authors demonstrated this principle by applying their methodology to two masked S-box designs following DOM. In short, SESYM replaces all gates of a DOM gadget by their WDDL counterparts (cf. Figure 4). Due to the underlying DRP, the transformed circuit guarantees to be glitch-free. Hence, the authors of SESYM eliminated the registers, and claimed that this does not affect the security since their only purpose is to stop the propagation of glitches.

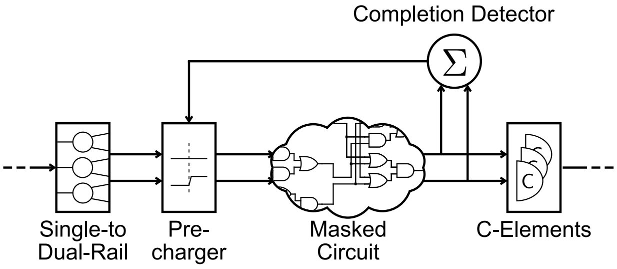
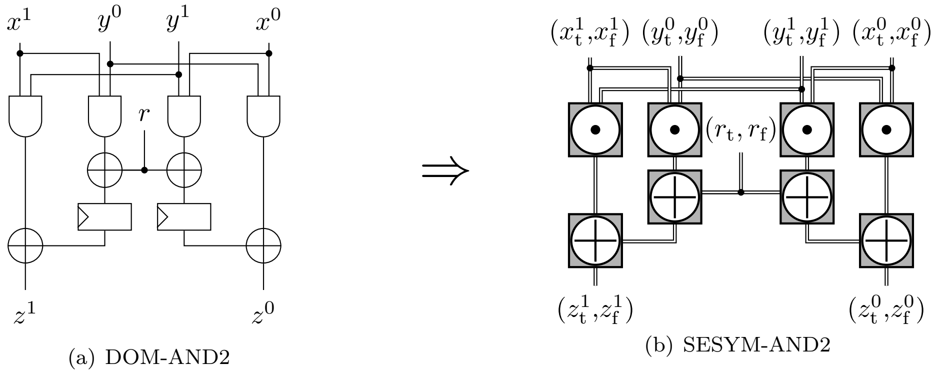


Figure 3: SESYM components [NGPM22].


x^1
y^0
y^1
x^0
r
z^1
z^0
$\Rightarrow$
$(x^1_\mathrm{t},x^1_\mathrm{f})$
$(y^0_\mathrm{t},y^0_\mathrm{f})$
$(y^1_\mathrm{t},y^1_\mathrm{f})$
$(x^0_\mathrm{t},x^0_\mathrm{f})$
$(r_\mathrm{t},r_\mathrm{f})$
$(z^1_\mathrm{t},z^1_\mathrm{f})$
$(z^0_\mathrm{t},z^0_\mathrm{f})$

(a) DOM-AND2

(b) SESYM-AND2

Figure 4: Conversion of a DOM-AND2 to SESYM.

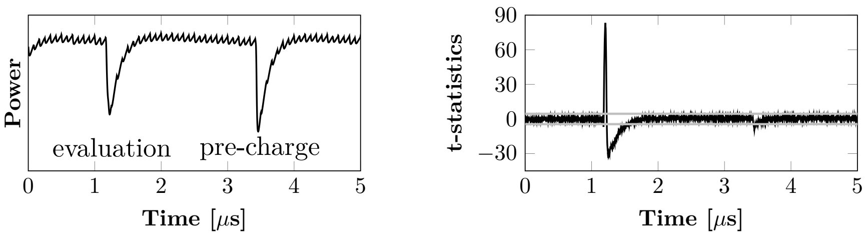
4 Experimental Analysis

Before theoretically evaluating these two design principles, we first demonstrate the result of experimental analyses conducted on an AES S-box implemented by each of these schemes. Below, we first give the specifications of the underling setup, before we provide detailed expression of each experiment individually.

4.1 Setup

Our entire analyses have been conducted on the AMD (Xilinx) Spartan-6 FPGA of an SCA-evaluation board SAKURA-G [SAK16]. We made use of a digital sampling oscilloscope to monitor the voltage drop over a 1 Ω shunt resistor placed on the Vdd path of the underlying Spartan-6 FPGA, which is proportional to the current passing through the FPGA, hence a fraction of its instantaneous power consumption. We collected such traces at a sampling rate of 500 MS/s while the FPGA was being supplied by a stable oscillator at a frequency of 6 MHz, hence minimizing the effect of adjacent clock cycles on each other via power traces. Further, we made use of a 10 dB AC amplifier to monitor the voltage drop of the aforementioned resistor with higher accuracy.

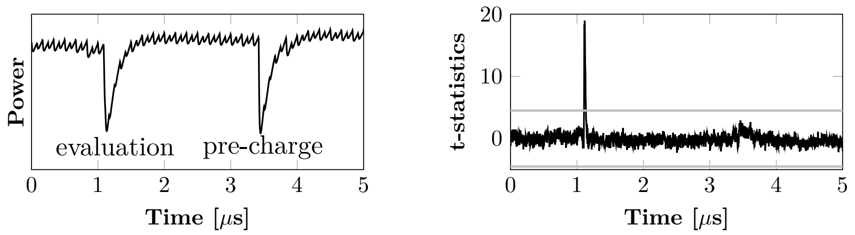
As the analysis technique, we followed the state of the art, and collected traces suitable for fixed-vs-random first-order leakage assessment t-test [SM15] to examine the existence of first-order leakages without conducting any attacks. Since all our designs under test are first-order masked, for every measurement the fixed or random input is given to the Spartan-6 FPGA in an independently and freshly masked form with two shares. When required, the fresh randomness (sometimes called online masks in various literature) are generated inside the Spartan-6 FPGA. To this end, we instantiated an individual 31-bit Linear Feedback Shift Register (LFSR) for each required fresh mask bit, which is seeded randomly at the power-up cycle of the Spartan-6 FPGA. This choice is also based on the



(a) Mean trace, over 100k traces

(b) t-test, over 1 million traces

Figure 5: SESYM AES S-box, original ASIC design.



(a) Mean trace, over 100k traces

(b) t-test, over 100 million traces

Figure 6: SESYM AES S-box, original FPGA design.

state of the art, particularly the highly efficient FPGA-specific implementation of such an LFSR used in [MMW18].

4.2 SESYM

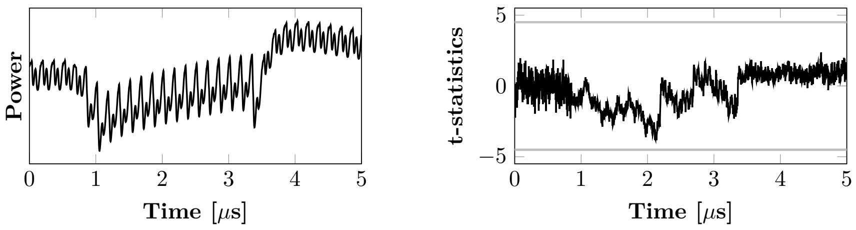
We first focus on the SESYM designs provided by the authors on a public Git repository[1].

ASIC-based design. As the initial design, we took the ASIC-based design, and instantiated a single module of such a masked AES S-box design. The architecture of the underlying S-box is known as Canright S-box [Can05], and requires 18 fresh mask bits. We made sure that every single gate is realized by a dedicated Look-Up Table (LUT). We further turned off all optimization features of the synthesizer (indeed FPGA toolchain Xilinx ISE) and kept the hierarchy of the design to avoid the synthesizer merging multiple gates into a LUT. Further, we do not restrict the placement and routing of the circuit and let the FPGA toolchain find the most suitable slices and resources.

The design contains a completion detector to initiate the pre-charge phase, but we controlled the evaluation and pre-charge phases and intentionally made a long delay between such phase changes to identify them in the power traces, as can also be seen in the mean trace shown in Figure 5(a). The corresponding t-test result using 100 million traces is shown in Figure 5(b), implying a highly detectable first-order leakage with less than 1 million traces.

FPGA-based design. We further examined the FPGA-based design that the authors also provided via Git[1]. We first noticed that the FPGA designs are entirely different to the ASIC ones. For example, the WDDL gadgets are realized by their LUT-based version without early evaluation and each LUT is also controlled by a global pre-charge signal,

[1] https://extgit.iaik.tugraz.at/sesys/self-synchronized-masking



(a) Mean trace, over 100k traces

(b) t-test, over 100 million traces

Figure 7: LMDPL AES S-box.

which forces all WDDL gadgets to move to the pre-charge phase simultaneously. Further, there is no completion detector, and the switch between pre-charge and evaluation phases is controlled by a Finite State Machine (FSM) according to the required operations of the AES. Although the design exhibits less leakage compared to the ASIC-based designs, its first-order leakage using 100 million traces is still clearly observable as shown in Figure 6(b).

In summary, our investigation into both designs implemented by the original authors of SESYM revealed significant first-order leakage in our experiments. Notably, the practical measurements conducted by the authors of SESYM did not exhibit such leakage. To elucidate this apparent discrepancy, it is crucial to clarify that, alongside the considerations of the applied measurement setup, the noise level of the target board, and environmental effects, the specific placement and routing of the circuit significantly impact the observed amount of leakageas both affect the propagation delays within the final design. If such propagation delays 1) differ enough to measure them in practice and 2) depend on sensitive data, we are able to detect significant information leakage in physical measurements. Our analyses demonstrate that this concern, is not merely theoretical but leads to substantial leakage in practical scenarios. It is plausible that the specific placement and routing employed in the original SESYM paper resulted in delays that either had no impact on security or were too small to be practically measured, unlike the case with the placement and routing of our designs. Nevertheless, considering that the placement and routing issues observed in our design can occur without manual intervention, it is imperative to underscore that the security of any masking scheme should be independent of its physical realization, encompassing placement and routing considerations.

4.3 LMDPL

We also conducted a similar experiment on an instance of the AES S-box made by LMDPL gadgets. The corresponding analysis result using 100 million traces is shown in Figure 7, which is consistent with the results the authors reported in [SBHM20], i.e., no detectable first-order leakage.

5 Problem in SESYM

We start by showing that glitch-free masked designs can still suffer from imbalanced delays, and how such delays can leak secret information. Therefore, we start with some definitions and examples to illustrate the origin of the problem.

Definition 7 (Delay)**.** A delay Δ_w of wire w is the time difference between 1) the moment when a combinational input transition occurs (such as a toggle at the register output or at

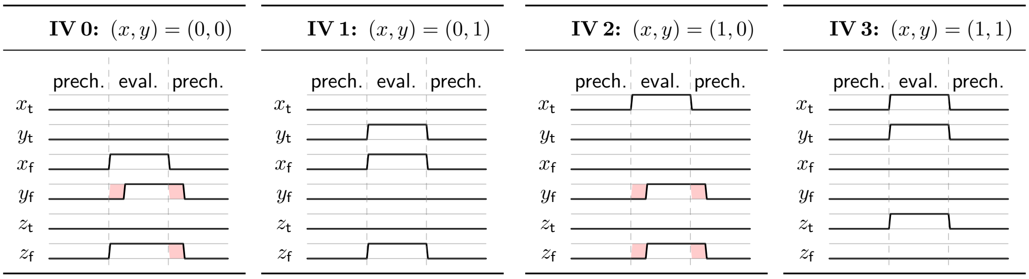


Figure 8: WDDL-AND2 timings, y_f delayed by $\Delta_{y_f} > \Delta_{y_t} = \Delta_{x_t} = \Delta_{x_f} = 0$.

a primary input), and 2) the moment when such a toggle is actually seen at the observation spot on w.

Without loss of generality, we define delays only of wires and ignore the time required for a gate to compute its outputs. This is because, time lags that occur at gates can be simply added to the delay of the gate's output wire. The observation spot is defined as the wire cross-section where a toggle is observed with a given delay. While the delay itself depends on the observation spot, we suppose that our assumptions hold for any arbitrary observation spot on the same wire, and thus do not focus on a specific observation spot for wire w. As outlined in Section 2, for circuits that employ DRP logic, we can make Definition 7 more explicit by introducing the following lemma.

Lemma 1. *During the pre-charge phase, Δ_w describes the delay between the start of the pre-charge phase and the transition of w from 1 to 0. During the evaluation phase, Δ_w describes the delay between the start of the evaluation phase and the transition of w from 0 to 1.*

Further, Lemma 1 implies that 1) Δ_w is only defined for transitions on w and that 2) Δ_w can differ depending on whether one observes the pre-charge phase or the evaluation phase. Hence, if a pre-charged wire w (driving zero) does not toggle during the evaluation phase, Δ_w cannot be observed. This also means that Δ_w is input-dependent as w may toggle only for certain primary inputs of the circuit. Initially, we focus on WDDL gadgets to visualize the underlying problem originating from delays.

Example 2 (WDDL-AND2 with delay)**.** Based on the sketch in Figure 1(a), let us consider a single WDDL-AND2 gadget with primary inputs x, y, and output z in dual-rail representation, i.e., (x_t, x_f), (y_t, y_f), and (z_t, z_f). Moreover, we assume that y_f is delayed compared to the other inputs. For simplicity, let us suppose that $\Delta_{y_f} > \Delta_{x_f}$ while $\Delta_{x_f} = \Delta_{x_t} = \Delta_{y_t} = 0$, i.e., we ignore all delays except Δ_{y_f}. Figure 8 visualizes the timing for all possible input vectors, denoted as IV 0 to IV 3.

Case 1 (Probe on z_t). Consider an adversary placing a probe, which continuously captures the signal carried by z_t during the evaluation phase. In the event that the probe records 1, it implies that both x_t and y_t are also equal to 1. Consequently, the adversary deduces that the specific scenario resulting from IV 3, where $(x, y) = (1, 1)$, has occurred. In contrast, if the probe records 0, it implies that either $x_t = 0$ or $y_t = 0$. Consequently, the adversary deduces that one of the IV 0 to IV 2 was processed but remains unable to determine whether it was specifically IV 0, IV 1, or IV 2.

Case 2 (Probe on z_f). Consider an adversary placing a probe, which continuously captures the signal carried by z_f during the evaluation phase. In the evaluation phase, the signal carried by z_f toggles from 0 to 1 in any of the cases IV 0 to IV 2, whereas this behavior is not exhibited for IV 3. Consequently, an adversary can deduce that one of the IV 0 to IV 2 was processed if the probe records a toggle. Furthermore, the adversary possesses the

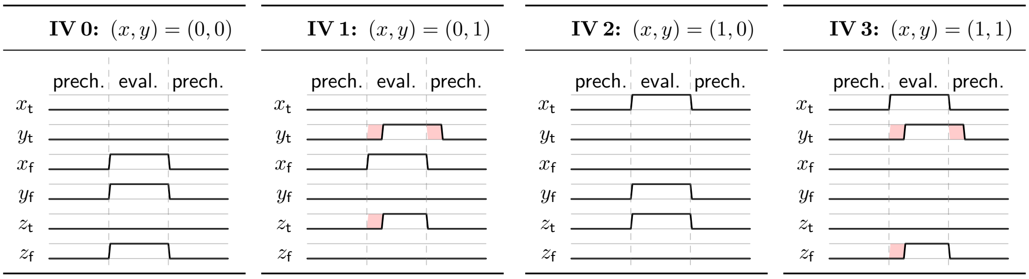


Figure 9: WDDL-XOR2 timings, y_t delayed by $\Delta_{y_\mathrm{t}} > \Delta_{y_\mathrm{f}} = \Delta_{x_\mathrm{t}} = \Delta_{x_\mathrm{f}} = 0$.

capability to distinguish between IV 2 and the other cases by analyzing the timing when the toggle occurs. If the toggle happens immediately at the start of the evaluation phase, i.e., $\Delta_{z_\mathrm{f}} = 0$, the adversary deduces that either IV 0 or IV 1 has occurred but remains unable to determine whether it was specifically IV 0 or IV 1. Conversely, if the toggle is delayed, i.e., $\Delta_{z_\mathrm{f}} = \Delta_{y_\mathrm{f}}$, the adversary conclude that IV 2 was processed.

Similarly, consider an adversary placing a probe, which continuously captures the signal carried by z_f during the subsequent pre-charge phase. In this phase, the signal carried by z_f toggles from 1 to 0 for the cases IV 0 to IV 2, whereas this behavior is not exhibited for IV 3. Consequently, an adversary can deduce that one of the IV to IV 2 has occurred if the probe records a toggle. Furthermore, the adversary possesses the capability to distinguish between IV 1 and the remaining cases by analyzing the timing of the toggle. If the toggle happens immediately at the start of the pre-charge phase, i.e., $\Delta_{z_\mathrm{f}} = 0$, the adversary detects that IV 1 was processed. Conversely, if the toggle is delayed, i.e., $\Delta_{z_\mathrm{f}} = \Delta_{y_\mathrm{f}}$, the adversary concludes that either IV 0 or IV 2 was processed without exactly knowing which one was the case.

Example 3 (WDDL-XOR2 with delay)**.** Following Figure 1(c), consider a WDDL-XOR2 gadget with primary inputs x, y, and output z in dual-rail representation, i.e., $(x_\mathrm{t}, x_\mathrm{f})$, $(y_\mathrm{t}, y_\mathrm{f})$, and $(z_\mathrm{t}, z_\mathrm{f})$. Moreover, we assume that y_t is delayed compared to the other inputs. For simplicity, we assume that $\Delta_{y_\mathrm{t}} > \Delta_{x_\mathrm{t}} = \Delta_{x_\mathrm{f}} = \Delta_{y_\mathrm{f}} = 0$. Figure 9 visualizes the timing for all possible input vectors.

Case 1 (Probe on z_t). Consider an adversary placing a probe, which continuously captures the signal carried by z_t during the evaluation phase. In the evaluation phase, the signal carried by z_t toggles from 0 to 1 in case of IV 1 and IV 2, whereas this is not exhibited for IV 0 and IV 3. Consequently, an adversary can deduce that either IV 1 or IV 2 has occurred if the probe records a toggle. Furthermore, the adversary possesses the capability to distinguish between IV 1 and IV 2 by analyzing the timing of the toggle occurrences. If the toggle happens immediately at the start of the evaluation phase, i.e., $\Delta_{z_\mathrm{t}} = 0$, the adversary can conclude that IV 2 was the case. However, if the toggle is delayed, i.e., $\Delta_{z_\mathrm{t}} = \Delta_{y_\mathrm{t}}$, the adversary deduces that IV 1 has taken place. In contrast, if the probe records 0, the adversary remains unable to determine whether it was specifically IV 0 or IV 3.

Case 2 (Probe on z_f). Consider an adversary placing a probe, which continuously captures the signal carried by z_f during the evaluation phase. In the evaluation phase, the signal carried by z_f toggles from 0 to 1 in case of IV 0 or IV 3, whereas this behavior is not exhibited in other cases. Consequently, the adversary can conclude that either IV 0 or IV 3 has occurred if the probe records a toggle. The adversary further possesses the capability to distinguish between IV 0 and IV 3 by analyzing when the toggle occurred. If the toggle happens immediately at the start of the evaluation phase, i.e., $\Delta_{z_\mathrm{f}} = 0$, the adversary recognizes that IV 0 has occurred. If the toggle is delayed, i.e., $\Delta_{z_\mathrm{f}} = \Delta_{y_\mathrm{t}}$, the adversary deduces that IV 3 has taken place. However, if the probe records 0, the adversary remains

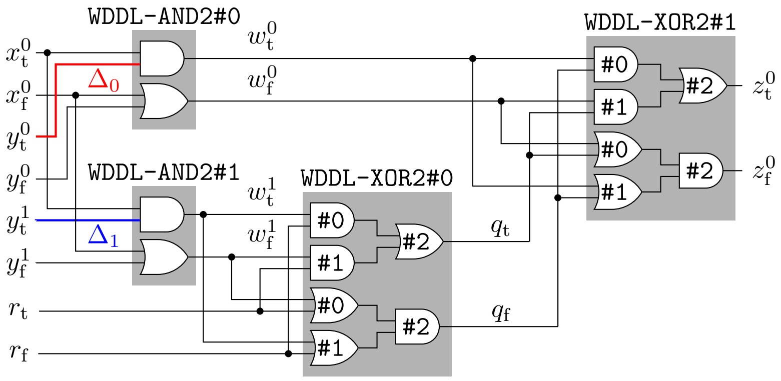


Figure 10: First-order SESYM-AND2 gadget (only first output share), red wire delayed by Δ_0, blue wire delayed by Δ_1.

unable to determine the exact case among IV 1 and IV 2.

Despite the assumption of the robust probing model that there is no probe-propagation through the combinational logic in a glitch-free circuit, both above examples demonstrate that information about the input of glitch-free DRP gates can be revealed by observing their outputs. This is due to different timing that is not accounted for by the robust probing model. We remark that this problem does not solely originate from the early propagation effect. For example, if WDDL-AND2-noEE is employed in Example 2 (cf. Figure 1(b)), the same information can be exploited when the delay of the rails are not balanced.

Next, we demonstrate that the information flow resulting from delays can lead to vulnerabilities in masked circuits that are assumed to be secure. To this end, we investigate a first-order SESYM-AND2 gadget (cf. Figure 4(b)) that takes x and y as inputs while the output is denoted by z. The gadget is first-order masked, i.e., $x = x^0 \oplus x^1$, $y = y^0 \oplus y^1$, and $z = z^0 \oplus z^1$, and operates on the dual-rail representation of each intermediate. Hence, each signal corresponds to a tuple as follows.

$$\begin{aligned} x^0 &:(x_t^0, x_f^0), & y^0 &:(y_t^0, y_f^0), & z^0 &:(z_t^0, z_f^0) \\ x^1 &:(x_t^1, x_f^1), & y^1 &:(y_t^1, y_f^1), & z^1 &:(z_t^1, z_f^1) \end{aligned}$$

Here, we focus on the computation of the dual-rail representation of z^0 as shown in Figure 10. We assume that y_t^0 and y_t^1 arrive asynchronously at SESYM-AND2's internal WDDL-AND2 gadgets. In particular, we assume that y_t^0 (resp. y_t^1) arrives with a delay of Δ_0 (resp. Δ_1) at `WDDL-AND2#0` (resp. `WDDL-AND2#1`). Thus, both delays contribute to the generation of the output share z^0. As before, we ignore all other delays, i.e., assume that they are zero.

Example 4 (SESYM-AND2)**.** Table 1 shows the effect of Δ_0 and Δ_1 when placing a probe on z_t^0 and observing its toggles during the evaluation phase. As y_t^0 (resp. y_t^1) is an input of the internal AND2 gate of `WDDL-AND2#0` (resp. `WDDL-AND2#1`), it holds that Δ_0 (resp. Δ_1) only propagates to the output w_t^0 (resp. w_t^1), if $x_t^0 = 1$ and $y_t^0 = 1$ (resp. $y_t^1 = 1$).

Case 1 (Only Δ_0 propagates). If Δ_0 goes through `WDDL-AND2#0` to w_t^0, w_t^0 also toggles with a delay of Δ_0. Further, `WDDL-XOR2#1` receives the delayed $w_t^0 = 1$ as an input to generate both rails z_t^0 and z_f^0. Depending on the value of q (output of `WDDL-XOR2#0`), the delay is either propagated to z_t^0 or z_f^0.

Table 1: Timing dependency of z_{t}^0 of SESYM-AND2 gadget with and without early evaluation (noEE), y_{t}^0 delayed by Δ_0 and y_{t}^1 by Δ_1.

							SESYM-AND2		**SESYM-AND2-noEE**	
$\boldsymbol{y}$	$\boldsymbol{x}$	$\boldsymbol{y^1}$	$\boldsymbol{y^0}$	$\boldsymbol{x^1}$	$\boldsymbol{x^0}$	$\boldsymbol{r}$	$\boldsymbol{\Delta_{z_{\mathrm{t}}^0}}$	$\boldsymbol{\overline{\Delta}_{\Delta_0<\Delta_1}}$	$\boldsymbol{\Delta_{z_{\mathrm{t}}^0}}$	$\boldsymbol{\overline{\Delta}_{\Delta_0<\Delta_1}}$
0	0	0	0	0	0	1	0	$\frac{\Delta_1}{4}$	0	$\frac{\Delta_1}{2}$
		0	0	1	1	1	0		0	
		1	1	0	0	1	0		$\max(\Delta_0, \Delta_1)$	
		1	1	1	1	1	$\max(\Delta_0, \Delta_1)$		$\max(\Delta_0, \Delta_1)$	
0	1	0	0	0	1	1	0	$\frac{\Delta_1}{4}$	0	$\frac{\Delta_1}{2}$
		0	0	1	0	1	0		0	
		1	1	0	1	1	$\max(\Delta_0, \Delta_1)$		$\max(\Delta_0, \Delta_1)$	
		1	1	1	0	1	0		$\max(\Delta_0, \Delta_1)$	
1	0	0	1	0	0	1	0	$\frac{\Delta_0+\Delta_1}{4}$	Δ_0	$\frac{\Delta_0+\Delta_1}{2}$
		0	1	1	1	0	Δ_0		Δ_0	
		1	0	0	0	1	0		Δ_1	
		1	0	1	1	0	Δ_1		Δ_1	
1	1	0	1	0	1	0	Δ_0	$\frac{\Delta_0+\Delta_1}{4}$	Δ_0	$\frac{\Delta_0+\Delta_1}{2}$
		0	1	1	0	1	0		Δ_0	
		1	0	0	1	0	Δ_1		Δ_1	
		1	0	1	0	1	0		Δ_1	

Case 2 (Only Δ_1 propagates). If Δ_1 propagates to w_{t}^1, w_{t}^1 toggles with a delay of Δ_1. Further, `WDDL-XOR2#0` receives the delayed $w_{\mathrm{t}}^1 = 1$ as an input to generate both rails q_{t} and q_{f}. Similar to *Case* 1, Δ_1 propagates either to q_{t} or q_{f} depending on the value of r.

Case 3 (Both Δ_0 and Δ_1 propagate). If w_{t}^0 and w_{t}^1 are delayed by Δ_0 and Δ_1, q reaches its valid dual-rail state with a delay of Δ_1. Hence, two delays affect the timing of `WDDL-XOR2#1`. In this case, it holds that $w_{\mathrm{t}}^0 = 1$ while q depends on r.

In Table 1, we excluded all indistinguishable input vectors, i.e., all inputs where z_{t}^0 does not toggle. In such cases, however, z_{f}^0 must toggle to keep a valid dual-rail state. Hence, all indistinguishable input vectors with respect to z_{t}^0 lead to a toggle on z_{f}^0. Hence, by placing a probe on z_{f}^0, the adversary would observe the same information but for other input vectors. In order to reflect first-order analysis, we further provided the average of delay observed at z_{t}^0, i.e., $\overline{\Delta}$, based on the unshared value of the inputs x and y, assuming that $\Delta_0 < \Delta_1$. In this example, it holds that $\overline{\Delta} = \frac{\Delta_1}{4}$ if $y = 0$, and $\overline{\Delta} = \frac{\Delta_0+\Delta_1}{4}$ if $y = 1$, i.e., a clear dependency of the timing to the unshared value of the input y.

noEE. We repeat Example 4 while exchanging all WDDL-AND2 gadgets with their alternative design WDDL-AND2-noEE (cf. Figure 1(b)) that avoids the early propagation effect as depicted in Figure 11.

Example 5 (SESYM-AND2-noEE)**.** The resulting effect of Δ_0 and Δ_1 on the output delay is also visualized in Table 1. The most important difference compared to Example 4 is that y_{t}^0 and y_{t}^1 affect the inverted output rail of `noEE-AND2#0` and `noEE-AND2#1` in Figure 11. Consequently, Δ_0 and Δ_1 can propagate to both output rails of w^0 and w^1. For the sake of simplicity, we do not repeat the cases with $w_{\mathrm{t}}^0 = 1$ or $w_{\mathrm{t}}^1 = 1$ as they are already covered in Example 4. In particular, we focus on cases where the delay propagates to w_{f}^0 or w_{f}^1.

Case 1 (Only Δ_0 propagates). We assume that Δ_0 propagates to w_{f}^0, i.e., $x_{\mathrm{f}}^0 = 1$ and $y_{\mathrm{t}}^0 = 1$ while `AND#2` of `noEE-AND2#0` evaluates to 1. Next, w_{f}^0 is forwarded to both output

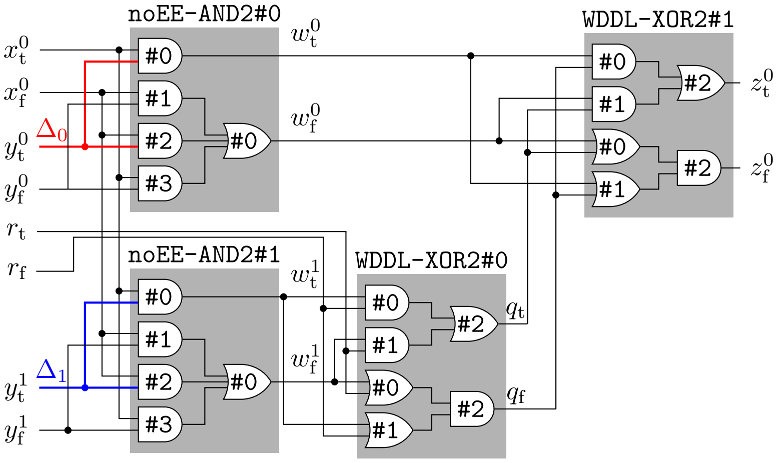


Figure 11: First-order SESYM-AND2-noEE gadget (only first output share), red wire delayed by Δ_0, blue wire delayed by Δ_1.

rails of WDDL-XOR2#1. Further, WDDL-XOR2#1 propagates Δ_0 either to z_t^0 or z_f^0. If $q = 1$, AND#1 evaluates to 1 with a delay of Δ_0, which is further propagated to z_t^0. If $q = 0$, OR#0 and OR#1 evaluate to 1, but OR#0 with a delay of Δ_0. Hence, AND#2 must wait for both signals before evaluating to 1. This leads to propagating Δ_0 to z_f^0.

Case 2 (Only Δ_1 propagates). We assume that Δ_1 propagates to w_f^1, i.e., $x_f^0 = 1$ and $y_t^1 = 1$ similar to noEE-AND2#0 in *Case* 1. Moreover, WDDL-XOR2#0 behaves similar to WDDL-XOR1#2 in *Case* 1 and propagates Δ_1 to either q_t or q_f depending on r. Finally, WDDL-XOR2#1 behaves similar to WDDL-XOR2#1 in *Case* 2 of Example 4. Hence, Δ_1 propagates to either z_t^0 or z_f^0.

Case 3 (Both Δ_0 and Δ_1 propagate). Δ_0 and Δ_1 propagate to w_f^0 and w_f^1 only if $x_f^0 = 1$, $y_t^0 = 1$, and $y_t^1 = 1$. As we know from *Case* 1 and *Case* 2, both delays are propagated to the final output. Hence, the longer delay must be considered.

The final delay observed at z_t^0 is also given in Table 1 implying that the time-of-evaluation of z_t^0 again depends on the unshared value of the input y. These examples demonstrate that the problem we have shown here does not originate from the early propagation effect of the underlying DRP logic.

We, further, replicated the steps outlined in Example 4 and Example 5 for the higher-order variants of the SESYM-AND2(-noEE) gadgets. Specifically, when examining a SESYM-AND2(-noEE) gadget operating on d shares per variable, it becomes evident that observing a single output share, either $z_t^{i\in\{0,\ldots,d-1\}}$ or $z_f^{i\in\{0,\ldots,d-1\}}$, with a probe is sufficient to discern the average delay based on the unshared value of input y. Hence, even higher-order variants of SESYM, operating on more than two shares per variable, provide no security against a (first-order) adversary with the ability to place a single probe.

6 Solution by LMDPL

With a particular focus on SESYM, we demonstrated examples in Section 5 where the propagation of delays could compromise the physical security of a masked circuit realized by a DRP logic. The primary concern arises from the lack of synchronization of intermediates through register stages in SESYM. Consequently, delays have the potential to traverse the entire circuit, thereby exposing information pertaining to multiple shares amalgamated during processing in the combinational circuit. It is crucial to reiterate that unintended information flow, even within DRP circuits, can only be mitigated by incorporating register stages. An illustrative example that employs DRP logic in conjunction with a single register stage is LMDPL. Subsequently, we delve into an examination of how delays propagate through a non-linear LMDPL gadget. This analysis aims to provide insights into the efficacy of LMDPL in managing delay propagation, effectively maintaining the physical security.

Example 6 (LMDPL-AND2)**.** Based on Figure 2, consider a non-linear LMDPL gadget with shared primary inputs (x^0, x^1) and (y^0, y^1) satisfying $x = x^0 \oplus x^1$ and $y = y^0 \oplus y^1$, and a shared primary output (z^0, z^1) with $z = z^0 \oplus z^1$. The mask table generation layer processes (x^0, y^0) together with a fresh mask r to compute z^0. The operation layer processes the dual-rail representation of the second input shares x^1 and y^1, and returns z^1 with

$$x^1 :(x^1_\mathrm{t}, x^1_\mathrm{f}), \qquad y^1 :(y^1_\mathrm{t}, y^1_\mathrm{f}), \qquad z^1 :(z^1_\mathrm{t}, z^1_\mathrm{f}).$$

The security proof given in [SBHM20] does already consider delays occurring in the evaluation layer. We repeat that LMDPL gadgets are restricted to only first-order security. Hence, we only consider adversaries with the ability to place a single probe. We remark that no signal processed within the mask table generation layer can leak secret information as they all depend solely on x^0, y^0, and r, i.e., an incomplete set of input shares. Therefore, we focus on the evaluation layer which processes the synchronized and blinded intermediates $t_0, \ldots, t_7$ (which depend on x^0, y^0, and r) together with (x^1_t, x^1_f) and (y^1_t, y^1_f).

Now, consider an adversary placing a probe on z^1_t which continuously captures the carried signal during the evaluation phase. As only one of $s_4, \ldots, s_7$ toggles from 0 to 1 during the evaluation phase (see Figure 2), it becomes evident that an adversary can distinguish which $s_{i \in \{4,\ldots,7\}}$ has toggled based on the observed delay Δ_{s_i} through the probe placed on z^1_t. However, detecting a toggle on s_i only permits the conclusion that all inputs of the corresponding AND gate computing s_i are 1 (see Figure 2), while at least one input of every other AND gate in the operation layer is 0. When detecting a particular $s_i = 1$, the adversary directly reveals x^1, y^1 and the corresponding $t_i = 1$. Since all t_4, …, t_7 are blinded by the fresh mask r, this does not lead to any information about the first input shares x^0 and y^0, hence no information about x and/or y. Further, the adversary can only differentiate between s_i being 0 or 1, but is unable to identify the input of specific AND gates generating $s_i = 0$. In other words, knowing that all other $s_i = 0$ provides no extra information about t_i. These observations are valid as well if the adversary places the probe on z^1_f.

To visualize these observations, we show the effect of arbitrary delays when observing the signal carried by z^1_t (resp. z^1_f) during the evaluation phase in Table 2. We assume that all input signals of the evaluation layer, i.e., x^1_t, x^1_f, y^1_t, y^1_f, t_0, …, t_7, are delayed by $\Delta_{x^1_\mathrm{t}}$, $\Delta_{x^1_\mathrm{f}}$, $\Delta_{y^1_\mathrm{t}}$, $\Delta_{y^1_\mathrm{f}}$, Δ_{t_0}, …, Δ_{t_7} respectively, while we ignore all other delays. Therefore, it holds that all Δ_{s_i} depend on the delay of the inputs of the evaluation layer. Exemplary, the following holds for those which are relevant to z^1_t.

$$\begin{aligned} \Delta_{s_4} &= \max(\Delta_{x^1_\mathrm{f}}, \Delta_{y^1_\mathrm{f}}, \Delta_{t_4}), & \Delta_{s_5} &= \max(\Delta_{x^1_\mathrm{t}}, \Delta_{y^1_\mathrm{f}}, \Delta_{t_5}), \\ \Delta_{s_6} &= \max(\Delta_{x^1_\mathrm{f}}, \Delta_{y^1_\mathrm{t}}, \Delta_{t_6}), & \Delta_{s_7} &= \max(\Delta_{x^1_\mathrm{t}}, \Delta_{y^1_\mathrm{t}}, \Delta_{t_7}). \end{aligned}$$

Similar to the analysis we performed for SESYM in Section 5, Table 2 only shows input vectors where z^1_t (resp. z^1_f) toggles. The average delay observed at z^1_t (resp. z^1_f) are also denoted by $\overline{\Delta}_{z^1_\mathrm{t}}$ (resp. $\overline{\Delta}_{z^1_\mathrm{f}}$) based on the unshared values of the inputs x and y. In contrast to the examples derived from SESYM, it holds that $\overline{\Delta}_{z^1_\mathrm{t}} = \frac{1}{4}\sum_{i=4}^{7} \Delta_{s_i}$ (resp. $\overline{\Delta}_{z^1_\mathrm{t}} = \frac{1}{4}\sum_{i=0}^{3} \Delta_{s_i}$) independent of x and y. Hence, no dependency between the timing and the unshared x and y is observed.

Table 2: Timing dependency of the second output share $(z_{\mathrm{t}}^1, z_{\mathrm{f}}^1)$ of LMDPL-AND2 gadget (only evaluation layer).

						LMDPL-AND2					
$\boldsymbol{y}$	$\boldsymbol{x}$	$\boldsymbol{y^1}$	$\boldsymbol{y^0}$	$\boldsymbol{x^1}$	$\boldsymbol{x^0}$	$\boldsymbol{r}$	$\boldsymbol{\Delta}_{z_{\mathrm{t}}^1}$	$\overline{\boldsymbol{\Delta}}_{z_{\mathrm{t}}^1}$	$\boldsymbol{r}$	$\boldsymbol{\Delta}_{z_{\mathrm{f}}^1}$	$\overline{\boldsymbol{\Delta}}_{z_{\mathrm{f}}^1}$
0	0	0	0	0	0	1	Δ_{s_4}	$\frac{1}{4}\sum_{i=4}^{7}\Delta_{s_i}$	0	Δ_{s_0}	$\frac{1}{4}\sum_{i=0}^{3}\Delta_{s_i}$
		0	0	1	1	1	Δ_{s_5}		0	Δ_{s_1}	
		1	1	0	0	1	Δ_{s_6}		0	Δ_{s_2}	
		1	1	1	1	1	Δ_{s_7}		0	Δ_{s_3}	
0	1	0	0	0	1	1	Δ_{s_4}	$\frac{1}{4}\sum_{i=4}^{7}\Delta_{s_i}$	0	Δ_{s_0}	$\frac{1}{4}\sum_{i=0}^{3}\Delta_{s_i}$
		0	0	1	0	1	Δ_{s_5}		0	Δ_{s_1}	
		1	1	0	1	1	Δ_{s_6}		0	Δ_{s_2}	
		1	1	1	0	1	Δ_{s_7}		0	Δ_{s_3}	
1	0	0	1	0	0	1	Δ_{s_4}	$\frac{1}{4}\sum_{i=4}^{7}\Delta_{s_i}$	0	Δ_{s_0}	$\frac{1}{4}\sum_{i=0}^{3}\Delta_{s_i}$
		0	1	1	1	1	Δ_{s_5}		0	Δ_{s_1}	
		1	0	0	0	1	Δ_{s_6}		0	Δ_{s_2}	
		1	0	1	1	1	Δ_{s_7}		0	Δ_{s_3}	
1	1	0	1	0	1	0	Δ_{s_4}	$\frac{1}{4}\sum_{i=4}^{7}\Delta_{s_i}$	1	Δ_{s_0}	$\frac{1}{4}\sum_{i=0}^{3}\Delta_{s_i}$
		0	1	1	0	0	Δ_{s_5}		1	Δ_{s_1}	
		1	0	0	1	0	Δ_{s_6}		1	Δ_{s_2}	
		1	0	1	0	0	Δ_{s_7}		1	Δ_{s_3}	

7 Experimental Validation

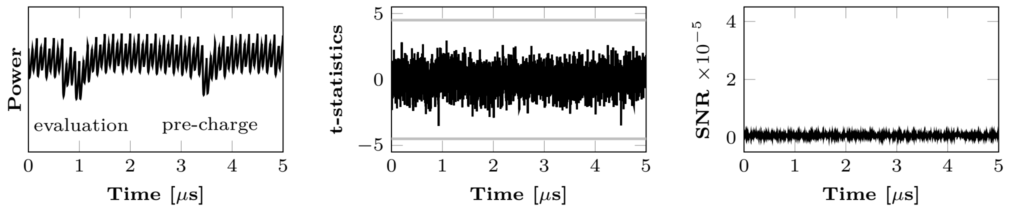
To experimentally confirm our theoretical observations with respect to insecurity of SESYM, we conducted a couple of FPGA-based analyses using the same setup expressed in Section 4.1.

As case study, we constructed the Keccak 5-bit χ function using SESYM multipliers. In short, it consists of five instances of such a multiplier and five masked XORs in DRP logic. The required inverters are just realized by swapping the complementary rails of one of the shares. Naturally, this design requires five fresh mask bits, each of which is individually associated to a SESYM multiplier.

The original SESYM multiplier design has been made for ASIC platforms. In order to be as close as possible to the original construction, similar to what explained in Section 4.2, we realized every single gate by a LUT, made sure that the design hierarchy is kept and the synthesizer does not trim, merge, or optimize any part of the circuit. Similar to the former experiments, we kept a long delay between the start of evaluation and pre-charge phases to better identify the source of leakage (if any). However, we did not limit the FPGA toolchain with respect to placement of logic cells and routing between them.

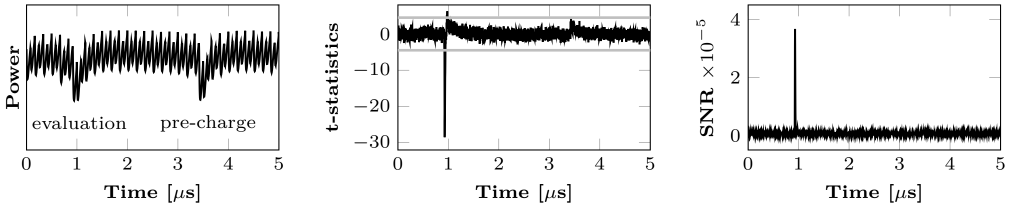
The corresponding mean trace and the t-test result using 100 million traces is shown in Figure 12(b), implying no detectable first-order leakage. We further used half of the collected traces (which belong to the group with random input) to estimate the Signal-to-Noise Ratio (SNR) [MOP07] based on the 5-bit unmasked input of the χ S-box. The resulting SNR curve is shown in Figure 12(c), also indicating not a visible dependency. We would like to highlight that the underlying design does not contain any cascaded SESYM multipliers, and has a very low power consumption.

In order to examine our claims given in Section 5, i.e., the effect of delay imbalances, we manually added delay elements (realized by cascaded inverters) to y_{t}^0 and y_{t}^1 of each WDDL-AND2 gadget of every SESYM multiplier (see the description and notation in



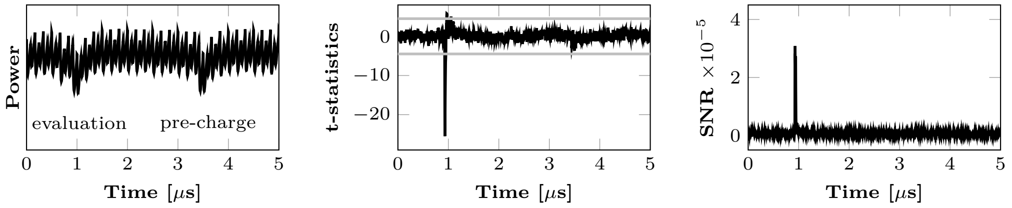
(a) Mean trace, over 100k traces (b) t-test, over 100 million traces (c) SNR, over 50 million traces

Figure 12: SESYM Keccak χ, uncontrolled placed and routed.



(a) Mean trace, over 100k traces (b) t-test, over 100 million traces (c) SNR, over 50 million traces

Figure 13: SESYM Keccak χ, with manually delayed signals.



(a) Mean trace, over 100k traces (b) t-test, over 100 million traces (c) SNR, over 50 million traces

Figure 14: SESYM Keccak χ, without early evaluation (noEE), with manually delayed signals.

Section 5). To this end, we instantiated 6 LUTs each realizing an inverter[2] in signal y_{t}^{0} and 10 LUTs in signal y_{t}^{1}. Repeating the same experiment led to the results shown in Figure 13, clearly showing the first-order leakage of the design as well as a visible data-dependency via SNR. Note that if the design is provably secure, its security should not depend on the delay of its internal signals, i.e., on the placement and routing. This can be seen as a counterexample for the claim of the authors in [NGPM22] that imbalanced dual rails do not have any effect on the security of SESYM circuits.

As stated above, we did our best to emulate the ASIC-based designs on the underlying FPGA. As given in Section 2, WDDL-AND2 gadgets have the early evaluation issue, which is also the case in the experiments explained above. The noEE version can mitigate this problem on FPGAs by instantiating one 5-to-2 LUT for the entire operation of a WDDL-AND2 gadget (see Section 2.3 and [MI14]). Hence, we repeated the last experiment by replacing every WDDL-AND2 gadget with WDDL-AND2-noEE (cf. Figure 1) while keeping the intentional imbalances between the aforementioned rails. This did not change the situation, and we obtained the same results as shown in Figure 14. This confirms our

[2] In FPGAs, LUTs can also be configured as buffer. This, however, does not have any effect on our experiment, since we used an even number of inverters.

claims and statements given in Section 5, that even in case of no early propagation, the imbalances between delay of rails would falsify the SESYM's security arguments.

8 Discussions

The results presented in Section 4 and Section 7 validate that the vulnerabilities identified in theoretical analyses of SESYM are indeed detectable in practical scenarios, underscoring the realistic threat posed by imbalanced delays in real-world implementations. As already explained in Section 4, a SESYM-protected design may not exhibit leakage with a certain placement and routing of its components but this does not guarantee its security for arbitrary circuits. In essence, the concrete physical realization of such a circuit significantly impacts its achieved security level. This complicates the way to verify the actual security level of the circuit, underscoring the imperative need for security proofs utilizing formal adversary models, such as the robust probing model. We acknowledge that formal security proofs tend to be overly conservative, accounting for all potential instances of physical defaults. This holds even for those that may never manifest in the physical circuit instance. However, formal security proofs remain the sole method for providing formal assurances regarding security, irrespective of the specific realization of the design. In the absence of a comprehensive security proof, there exists a risk that a design presumed to be secure and validated through experiments may – in practice – exhibit leakage. A recent example for such a case is discussed in [LMMS23].

In addition to practical measurements, the authors of SESYM formally verified the probing security of their approach using COCO [GHP$^+$21] omitting glitch-based probe propagation due to the absence of glitches. Apparently it was observed that COCO reports no leakage, in contrast to our theoretical findings outlined in Section 5. The authors of SESYM manually integrated the applied WDDL gadgets into COCO and constructed their respective correlation sets without considering glitches (see Section 5.2 of [NGPM22]). Each correlation set in this process only encompasses a constant value and the primary gate output, implying no probe propagation at all. This approach aligns with an evaluation under the standard d-probing model, which neglects any physical defaults. In our theoretical exploration, we demonstrated that delays can lead to probe propagation, highlighting the insufficiency of solely relying on the d-probing model. Theoretically, this implies the necessity to prove the probing security of glitch-free circuits akin to circuits with glitches, involving a comprehensive analysis of probe propagation. However, such a proof could be overly conservative as it overlooks the glitch-free nature of the circuit. For instance, LMDPL cannot be proven secure under the glitch-extended probing model, despite our observations indicating its security. In other words, we confirmed that LMDPL stays secure in the presence of delays while assuming that the probes are extended irrespective of the glitch-free nature of the circuit would lead to leakage as also shown in [SBHM20]. Consequently, the security assessment of DRP circuits under the robust d-probing model may not provide accurate insights, as it considers glitches which never violate the security of glitch-free circuits. Therefore, it is imperative to adapt the existing robust probing model to accurately capture the leakage behavior of glitch-free circuits. This adjustment represents a promising future research direction for more precise and reliable security evaluations.

The decisive question revolves around whether the necessity for formal security proofs, coupled with our observation that even glitch-free circuits exhibit some form of information propagation, allows any scope for fully asynchronous circuits without any clock-driven synchronization. We remark that realizing provably secure fully asynchronous circuits mandates that each gate attains a data-independent time-of-evaluation, and the delay of dual rails must be impeccably balanced, an undertaking that appears challenging. Consequently, while certain implementations made by such schemes may achieve practical security, the formal proof of their security seems unattainable. Nevertheless, the LMDPL concept shows that the application of DRP logic can lead to a provable secure low-latency solution. However, it necessitates a single register stage, rather than presenting a fully asynchronous self-timed solution operating within a single clock cycle.

9 Conclusions

In summary, we demonstrated that – in addition to glitches – signal delays such as routing imbalances can also compromise the security of masked hardware designs. We presented several examples to illustrate how imbalanced delays can propagate through combinational circuits and expose sensitive information even if the circuit is glitch-free. Moreover, we investigated the security of LMDPL and SESYM in theory and practice revealing security flaws in SESYM. We made use of imbalanced delays as a potential security flaw, which is known in the context of DRP circuits, criticising the security of power-equalizing hiding countermeasures [SS06]. We have shown that such a concept may compromise the security of glitch-free masked circuits as well, what is not covered by state-of-the-art models and proofs. Note that since some masking schemes like Threshold Implementation (TI) [NRR06] keep their security guarantees for all possible occurrences of glitches, they implicitly cover imbalanced delays as well. The authors of LMDPL incorporated delays as part of their theoretical discussions. However, the authors of SESYM formally verified the probing security of their approach through COCO [GHP+21] while ignoring glitch-based probe propagations due to the absence of glitches. Nevertheless, the probes are still propagated due to imbalanced delays, which COCO or other verification tools do not cover.

Acknowledgments

The work described in this paper has been supported in part by the German Research Foundation (DFG) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972, and through the projects 435264177 (SAUBER) and 456967092 (SecFShare).

References

[AG01] Mehdi-Laurent Akkar and Christophe Giraud. An Implementation of DES and AES, Secure against Some Attacks. In *CHES 2001*, volume 2162 of *LNCS*, pages 309–318. Springer, 2001.

[BGF+10] Shivam Bhasin, Sylvain Guilley, Florent Flament, Nidhal Selmane, and Jean-Luc Danger. Countering early evaluation: an approach towards robust dual-rail precharge logic. In *WESS 2010*, page 6. ACM, 2010.

[Can05] David Canright. A Very Compact S-Box for AES. In *CHES 2005*, volume 3659 of *LNCS*, pages 441–455. Springer, 2005.

[CBG+17] Thomas De Cnudde, Begül Bilgin, Benedikt Gierlichs, Ventzislav Nikov, Svetla Nikova, and Vincent Rijmen. Does Coupling Affect the Security of Masked Implementations? In *COSADE 2017*, volume 10348 of *LNCS*, pages 1–18. Springer, 2017.

[CGLS21] Gaëtan Cassiers, Benjamin Grégoire, Itamar Levi, and François-Xavier Standaert. Hardware Private Circuits: From Trivial Composition to Full Verification. *IEEE Trans. Comp.*, 70(10):1677–1690, 2021.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 398–412. Springer, 1999.

[CPRR13] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-Order Side Channel Security and Mask Refreshing. In *FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 2013.

[FGP$^+$18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable Masking Schemes in the Presence of Physical Defaults & the Robust Probing Model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):89–120, 2018.

[GHP$^+$21] Barbara Gigerl, Vedad Hadzic, Robert Primas, Stefan Mangard, and Roderick Bloem. Coco: Co-Design and Co-Verification of Masked Software Implementations on CPUs. In *USENIX Security 2021*, pages 1469–1468. USENIX Association, 2021.

[GMK16] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. In *TIS @ CCS 2016*, page 3. ACM, 2016.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic Analysis: Concrete Results. In *CHES 2001*, volume 2162 of *LNCS*, pages 251–261. Springer, 2001.

[GSST07] Jacopo Giorgetti, Giuseppe Scotti, Andrea Simonetti, and Alessandro Trifiletti. Analysis of data dependence of leakage current in CMOS cryptographic hardware. In *ACM Great Lakes Symposium on VLSI 2007*, pages 78–83. ACM, 2007.

[GST14] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. In *CRYPTO 2014*, volume 8616 of *Lecture Notes in Computer Science*, pages 444–461. Springer, 2014.

[HS13] Michael Hutter and Jörn-Marc Schmidt. The Temperature Side Channel and Heating Fault Attacks. In *CARDIS 2013*, volume 8419 of *Lecture Notes in Computer Science*, pages 219–235. Springer, 2013.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, 2003.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.

[KMMS22] David Knichel, Amir Moradi, Nicolai Müller, and Pascal Sasdrich. Automated Generation of Masked Hardware. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):589–629, 2022.

[Koc96] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO 1996*, volume 1109 of *LNCS*, pages 104–113. Springer, 1996.

[LMMS23] Daniel Lammers, Amir Moradi, Nicolai Müller, and Aein Rezaei Shahmirzadi. A thorough evaluation of RAMBAM. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *SIGSAC CCS 2023*, pages 1705–1717. ACM, 2023.

[LMW14] Andrew J. Leiserson, Mark E. Marson, and Megan A. Wachs. Gate-level masking under a path-based leakage metric. In *CHES 2014*, volume 8731 of *LNCS*, pages 580–597. Springer, 2014.

[MI14] Amir Moradi and Vincent Immler. Early propagation and imbalanced routing, how to diminish in fpgas. In *CHES 2014*, volume 8731 of *LNCS*, pages 598–615. Springer, 2014.

[MKEP12] Amir Moradi, Mario Kirschbaum, Thomas Eisenbarth, and Christof Paar. Masked Dual-Rail Precharge Logic Encounters State-of-the-Art Power Analysis Methods. *IEEE Trans. Very Large Scale Integr. Syst.*, 20(9):1578–1589, 2012.

[MMW18] Lauren De Meyer, Amir Moradi, and Felix Wegener. Spin Me Right Round Rotational Symmetry for FPGA-Specific AES. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):596–626, 2018.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[MPG05] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 351–365. Springer, 2005.

[MPO05] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully Attacking Masked AES Hardware Implementations. In *CHES*, volume 3659 of *LNCS*, pages 157–171. Springer, 2005.

[NGPM22] Rishub Nagpal, Barbara Gigerl, Robert Primas, and Stefan Mangard. Riding the waves towards generic single-cycle masking in hardware. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(4):693–717, 2022. Git repository: https://extgit.iaik.tugraz.at/sesys/self-synchronized-masking.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold Implementations Against Side-Channel Attacks and Glitches. In *ICICS 2006*, volume 4307 of *LNCS*, pages 529–545. Springer, 2006.

[OMPR05] Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller, and Vincent Rijmen. A Side-Channel Analysis Resistant Description of the AES S-Box. In *FSE 2005*, volume 3557 of *LNCS*, pages 413–423. Springer, 2005.

[PKZM07] Thomas Popp, Mario Kirschbaum, Thomas Zefferer, and Stefan Mangard. Evaluation of the Masked Logic Style MDPL on a Prototype Chip. In *CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2007.

[PM05] Thomas Popp and Stefan Mangard. Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. In *CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 2005.

[SAK16] SAKURA. Side-channel Attack User Reference Architecture. http://satoh.cs.uec.ac.jp/SAKURA/index.html, 2016.

[SBHM20] Pascal Sasdrich, Begül Bilgin, Michael Hutter, and Mark E. Marson. Low-latency hardware masking with application to AES. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(2):300–326, 2020.

[Sha79] Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979.

[SM15] Tobias Schneider and Amir Moradi. Leakage Assessment Methodology - A Clear Roadmap for Side-Channel Evaluations. In *CHES 2015*, volume 9293 of *LNCS*, pages 495–513. Springer, 2015.

[SS06] Daisuke Suzuki and Minoru Saeki. Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. In *CHES 2006*, volume 4249 of *LNCS*, pages 255–269. Springer, 2006.

[TV04a] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In *DATE 2004*, pages 246–251. IEEE Computer Society, 2004.

[TV04b] Kris Tiri and Ingrid Verbauwhede. Place and route for secure standard cell design. In *CARDIS 2004*, volume 153 of *IFIP*, pages 143–158. Kluwer/Springer, 2004.