Threshold Implementations Are Not Provably Secure Against Fault Sensitivity Analysis

Jeroen Delvaux

Open Security Research (OSR), Room 29–31, Floor 8, Building 12B,
Shenzhen Bay Tech-Eco Park, 518000 Shenzhen, China
jeroen.delvaux@osr-tech.com

Abstract. In an article presented at FDTC 2018, Arribas, De Cnudde, and Šijačić prove under mild conditions that *threshold implementations* (TIs) are secure against *fault sensitivity analysis* (FSA). Later in 2018, in the PhD thesis of De Cnudde, additional assumptions were imposed to provably withstand FSA, thereby increasing the required number of random bits. We point out that even under the latter, stronger conditions, the proof remains incorrect.

Keywords: Threshold Implementations · Fault Sensitivity Analysis

1 Introduction

Even for a cryptographic algorithm that is unbreakable in a purely mathematical sense, its implementation on an electronic device might be vulnerable to physical attacks. Measurable physical quantities leaked by a device, such as its power consumption and its electromagnetic emissions, depend on the secret intermediate variables that are being processed. To impede secrets from being retrieved through these *side channels*, *masking schemes* randomize computations such that leaked physical signals are independent of internal secrets up to a certain *statistical moment*, which is referred to as the *order*. *Threshold implementations* (TI) are a popular masking method as few assumptions about the hardware are made in their security proofs.

An attacker, however, is not limited to being a passive observer, and might actively induce faults into a computation, *e.g.*, by manipulating the clock signal or the supply voltage. As faulty outputs are exploitable through, *e.g.*, *differential fault analysis* (DFA) [2], cryptographic algorithms are often implemented in a redundant way such that faulty outputs can be detected and subsequently suppressed. In its simplest form, the algorithm is run twice; different outcomes imply that a fault must have occurred. Several types of *fault attacks*, however, including a *statistical ineffective fault attack* (SIFA) [7] and *fault sensitivity analysis* (FSA) [11], do not necessarily require faulty outputs, and might only require knowledge of whether the outputs are faulty or not. Ironically, redundant implementations with output suppression conveniently provide this one bit of information to the attacker. In an attempt to *kill two birds with one stone*[1], Arribas *et al.* [1] prove that TIs are secure against FSA. One coauthor, De Cnudde [6], later imposed additional conditions for the proof to hold, thereby significantly increasing a TI's intake of random bits.

[1] To be interpret in the idiom's symbolic sense. We do not condone cruelty to animals.

1.1 Contribution

We argue that the FSA-resistance proof, both in its original form by Arribas *et al.* [1] and in its revised from by De Cnudde [6], is fundamentally flawed. To strengthen our claim, we specify instances of TIs that succumb to FSA. We also point out that both versions of the proof [1, 6] are riddled with physical misassumptions, all of which happen to be inconsequential to the correctness of the proof.

1.2 Structure

The remainder of this article is structured as follows. Section 2 provides preliminaries. Section 3 refutes the FSA-resistance proof. Section 4 concludes this work.

2 Preliminaries

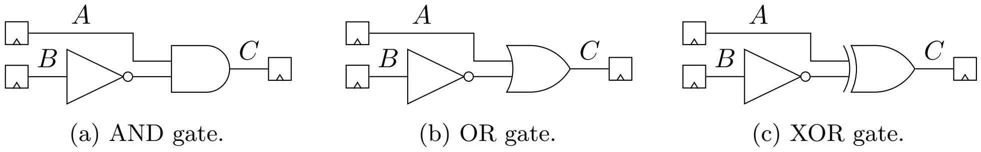
Section 2.1 introduces the notation. Sections 2.2 to 2.4 introduce the fundamentals of FSA, TIs, and the FSA-resistance proof.

2.1 Notation

Variables and constants are denoted by characters from the Latin and Greek alphabet respectively. A random variable is denoted by an uppercase character, e.g., X. Binary vectors are denoted by a bold-faced, lowercase character, e.g., $\mathbf{x}$. The all-zeros vector is denoted by $\mathbf{0}$; the all-ones vector is denoted by $\mathbf{1}$. The set of all λ-bit vectors is denoted by $\{0,1\}^\lambda$.

2.2 Fault Sensitivity Analysis

The propagation delay of a function G implemented as *combinational logic* depends on the value of its input data. Li *et al.* [11] illustrate this data dependency for the three types of two-input gates shown in Fig. 1, while assuming that gate propagation delays can accurately be described by a single parameter. Input B arrives later than input A due to, for example, an additional inverter. If for the AND gate, $A = 0$, the output quickly settles to $C = 0$, whereas for $A = 1$, more time is needed until the output $C = B$ is determined. This difference is formalized in Eq. (1), and similarly for the OR gate. The XOR gate does not exhibit any data dependency: the delay is a constant $D = D_{\text{INV}} + D_{\text{XOR}}$.



(a) AND gate. (b) OR gate. (c) XOR gate.

Figure 1: AND and OR gates induce data-dependent delays; XOR gates do not.

$$D = \begin{cases} D_{\text{AND}} & \text{, if } A = 0 \\ D_{\text{NOT}} + D_{\text{AND}} & \text{, otherwise,} \end{cases} \qquad D = \begin{cases} D_{\text{OR}} & \text{, if } A = 1 \\ D_{\text{NOT}} + D_{\text{OR}} & \text{, otherwise.} \end{cases} \tag{1}$$

Also in larger circuits, such as a *substition box* (S-box) of a symmetric-key cipher, the propagation delay for settling each output bit depends on the values of the input bits. Therefore, the sensitivity to register *setup-time* violations is data-dependent. An attacker can measure this sensitivity for a given input by gradually increasing the intensity level of a fault injection tool until faulty behavior appears. For example, the time offset between two consecutive rising edges in a clock signal can be progressively decreased. Alternatively, the clock signal is unmodified, but all propagation delays are increased, either by increasing the temperature or by decreasing the supply voltage.

Faulty outputs are not necessarily required for the attack to succeed, but can significantly improve its spatial locality. For example, for a layer of parallel S-boxes, the faultiness of an individual S-box output can be assessed rather than the faultiness of the complete ciphertext. In its original form, the attack requires a mathematical model of the data-dependent fault sensitivity, *i.e.*, knowledge of the circuit or even the layout is required. Two variations of FSA avoid this burden. First, for an S-box-like subcircuit that receives two identical inputs in subsequent clock cycles, the propagation delay is zero in the second clock cycle, *i.e.*, occurrences of this exceptionally low fault sensitivity are easy to spot [10, 12]. Second, it suffices that identical subcircuits, *e.g.*, two S-boxes, have similar data-dependent fault sensitivities such that subkey relations can be established by finding collisions [11, 13].

2.3 Threshold Implementations

In additive Boolean masking schemes, secrets $\mathbf{x} \in \{0,1\}^\lambda$ are randomly and uniformly split into σ shares according to Definition 1, thereby achieving the provably property given in Lemma 1 [14]. One way to meet Definition 1 is to first select $(\sigma - 1)$ masks $\mathbf{m}$ randomly, uniformly, and independently from $\{0,1\}^\lambda$, followed by the computation in Eq. (3).

Definition 1 (Uniformity)**.** A secret $\mathbf{x} \in \{0,1\}^\lambda$ is randomly and uniformly split into σ shares, *i.e.*, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\sigma \in \{0,1\}^\lambda$, if and only if the *probability mass function* (PMF) of $(X_1, X_2, \cdots, X_\sigma)$ given X is given in (2).

$$\mathbb{P}\big((X_1, \cdots, X_\sigma) = (\mathbf{x}_1, \cdots, \mathbf{x}_\sigma) \mid X = \mathbf{x}\big) = \begin{cases} 2^{-\lambda(\sigma-1)} & \text{, if } \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_\sigma = \mathbf{x} \\ 0 & \text{, otherwise.} \end{cases} \tag{2}$$

Lemma 1 (Subset of Shares)**.** *For a secret X that is randomly and uniformly split into σ shares according to Definition 1, it holds that any tuple of at most $\sigma - 1$ shares is independent of X.*

$$\mathbf{x}_1 = \mathbf{m}_1, \mathbf{x}_2 = \mathbf{m}_2, \cdots, \mathbf{x}_{\sigma-1} = \mathbf{m}_{\sigma-1}, \mathbf{x}_\sigma = \mathbf{x} \oplus \mathbf{m}_1 \oplus \mathbf{m}_2 \oplus \cdots \oplus \mathbf{m}_{\sigma-1}. \tag{3}$$

For a function of the form $\mathsf{G} : \{0,1\}^\lambda \to \{0,1\}^\eta$, a TI [14, 3] of G transforms σ_{in} shares of G's input $\mathbf{x}$ into σ_{out} shares of G's output $\mathbf{y} \triangleq \mathsf{G}(\mathbf{x})$, and consists of σ_{out} *component functions* $\mathsf{G}_i : \{0,1\}^\lambda \times \{0,1\}^\lambda \times \cdots \times \{0,1\}^\lambda \to \{0,1\}^\eta$ such that the correctness and δ^{th}-order incompleteness requirements in Definition 2 and Definition 3 respectively are met. A TI where $\delta = 1$ and $\sigma_{\text{in}} = \sigma_{\text{out}} = 3$, which Arribas *et al.* [1, 6] use as an example to develop their FSA-resistance proof, may involve computations $\mathbf{y}_1 \triangleq \mathsf{G}_1(\mathbf{x}_2, \mathbf{x}_3)$, $\mathbf{y}_2 \triangleq \mathsf{G}_2(\mathbf{x}_1, \mathbf{x}_3)$, and $\mathbf{y}_3 \triangleq \mathsf{G}_3(\mathbf{x}_1, \mathbf{x}_2)$ as shown in Fig. 2. As implied by Theorem 1, such a TI is only guaranteed to exist if G's algebraic degree $\tau \leq 2$. For affine functions G, *i.e.*, $\tau = 1$, TI are trivially constructed by setting $\sigma_{\text{in}} = \sigma_{\text{out}} = 2$ and letting $\mathsf{G}_1(\mathbf{x}_1) \triangleq \mathsf{G}(\mathbf{x}_1)$ and $\mathsf{G}_2(\mathbf{x}_2) \triangleq \mathsf{G}(\mathbf{x}_2)$.

Definition 2 (Correctness)**.** The list of component functions, *i.e.*, $\mathsf{G}_1, \mathsf{G}_2, \cdots, \mathsf{G}_{\sigma_{\text{out}}}$, is correct if and only if it holds for all tuples of shares $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\sigma_{\text{in}}}) \in \{0,1\}^\lambda \times \{0,1\}^\lambda \times \cdots \times \{0,1\}^\lambda$ that $\mathsf{G}_1(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\sigma_{\text{in}}}) \oplus \mathsf{G}_2(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\sigma_{\text{in}}}) \oplus \cdots \oplus \mathsf{G}_{\sigma_{\text{out}}}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\sigma_{\text{in}}}) = \mathsf{G}(\mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_{\sigma_{\text{in}}}) = \mathsf{G}(\mathbf{x})$.

Definition 3 (Incompleteness)**.** The list of component functions, *i.e.*, $\mathsf{G}_1, \mathsf{G}_2, \cdots, \mathsf{G}_{\sigma_{\text{out}}}$, is incomplete to the δ^{th} order if and only if any out of $\binom{\sigma_{\text{out}}}{\delta}$ combinations of component functions G_i does not depend on at least one input share X_i.

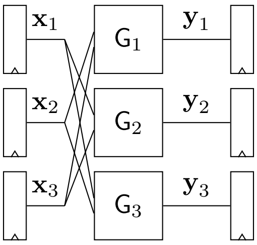


Figure 2: First-order TI.

Theorem 1 (Number of Shares)**.** *For any function* G *having algebraic degree* $\tau \in \mathbb{N}_0$ *and for any security order* $\delta \in \mathbb{N}_0$*, there exist a TI having* $\sigma_{\text{in}} \geq \tau\,\delta + 1$ *input shares and* $\sigma_{\text{out}} \geq \binom{\sigma_{\text{in}}}{\tau}$ *output shares.*

For a TI where the component functions G_i are evaluated in parallel, as previously shown in Fig. 2, Theorem 2 [3] implies that the desired security level against side-channel attacks can only be achieved if the physical leakages of all G_i's are independent of another. In practice, this assumption might only be approximately correct [4]: electric wires belonging to different G_i's can exhibit capacitive or inductive couplings, for example. Nevertheless, compared to preexisting masking schemes, in which similar independency assumptions are made, TIs have the advantage of not imposing constraints on the internals of each individual G_i. Most notably, the scheme tolerates *glitches* [15], *i.e.*, imbalanced propagation delays may cause circuit nodes to exhibit multiple transitions in a single clock cycle before settling to the correct logic level.

Theorem 2 (Security of Parallelized TI)**.** *For a TI having order* $\delta \in \mathbb{N}_0$ *and operating on a secret* X *that is randomly and uniformly split into* σ_{in} *shares, it holds for any physically leaked variable of the form* $L = L_{\mathsf{G}_1} + L_{\mathsf{G}_2} + \cdots + L_{\mathsf{G}_{\sigma_{\text{out}}}}$ *that the* δ^{th} *statistical moment of* L *is independent of* X*.*

For a composition of two functions, $\mathsf{G} \circ \mathsf{F}$, TIs of G and F cannot simply be put in series. First, a register layer should separate both TIs to avoid violating the incompleteness requirement given in Definition 3. Second, to ensure that Theorem 2 applies to the TI of G, the output shares of the TI of F should be uniform according to Definition 1. This requirement can be met either through imposing additional design constraints on the component functions F_i or through a form of remasking [5]. Note that block ciphers can be understood as a composition of identical round functions, *i.e.*, $\mathsf{G} \circ \mathsf{G} \circ \ldots \circ \mathsf{G}$.

2.4 FSA-Resistance Proof

The original and the revised version of the FSA-resistance proof are reviewed below.

2.4.1 Original Version

Despite a demonstration by Moradi *et al.* [13] that several masking schemes are vulnerable to FSA, Arribas *et al.* [1] argue that TIs are provably secure thanks to their incompleteness property. The proof further considers an isolated TI, given that standard rules for function composition still apply. Assuming all TI-inherent requirements are met, which includes uniformity of the input shares but excludes uniformity of the output shares, Assumption 1 and Assumption 2 are made to resist FSA. Assumption 1 is supplemented with a summary of the original FSA by Li *et al.* [11] where the fault intensity is gradually increased until a fault occurs.

Assumption 1. *FSA relies on the measurement of propagation delays.*

Assumption 2. *The component functions $\mathsf{G}_1, \mathsf{G}_2, \cdots, \mathsf{G}_\sigma$ operate in parallel and independently of one another, as depicted in Fig. 2.*

The security proof is elaborated for a TI of order $\delta = 1$ that operates on $\sigma_{\text{in}} = \sigma_{\text{out}} = 3$ shares, but can trivially be generalized to cover other parameter values. Upon adopting the same view of data-dependent propagation delays as Li *et al.* [11], let $D_{\mathsf{G}_1}(X_2, X_3, Y_1)$, $D_{\mathsf{G}_2}(X_1, X_3, Y_2)$, and $D_{\mathsf{G}_3}(X_1, X_2, Y_3)$ denote the largest propagation delays in their respective component functions for given input shares (X_1, X_2, X_3) and given output shares (Y_1, Y_2, Y_3). In this notation, the authors implicitly assume that (i) the register storing (X_1, X_2, X_3) has previously been reset to a known constant, *e.g.*, $(\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3) \triangleq (\mathbf{0}, \mathbf{0}, \mathbf{0})$, and (ii) the output shares (Y_1, Y_2, Y_3) are the result of evaluating this constant, *i.e.*, $Y_1 \triangleq \mathsf{G}_1(\boldsymbol{\chi}_2, \boldsymbol{\chi}_3)$, given that the inclusion of $Y_1 \triangleq \mathsf{G}_1(X_2, X_3)$ would be redundant, and similarly for Y_2 and Y_3. Without loss of generality, it can be assumed that $D_{\mathsf{G}_1} \geq D_{\mathsf{G}_2} \geq D_{\mathsf{G}_3}$. As G_1 is the first component function to produce a faulty output share, G_2 and G_3 do not affect the fault sensitivity of the TI as a whole. Globally, the attacker knows whether or not G_1 failed, but it cannot be measured whether or not G_2 and G_3 failed as well. As G_1 is independent of input share X_1, Lemma 1 implies that the attacker obtains no information about $X = X_1 \oplus X_2 \oplus X_3$.

2.4.2 Revised Version

In the revised version, De Cnudde [6] additionally imposes Assumption 3 and Assumption 4, using block-cipher terminology. A specifically mentioned scenario satisfying Assumption 3 is redundancy-based fault detection with output suppression, used as a countermeasure against DFA. Conceivably, an attacker only knows whether the output is correct or not. Assumption 4 precludes the aforementioned variation of FSA where identical inputs in subsequent clock cycles are spotted [10, 12]. The proof itself remains the same, except for notational differences that make D_{G_1}, D_{G_2}, and D_{G_3} dependent on the randomly selected reset value. The exact nature of this dependency is underspecified.

Assumption 3. *The attacker does not exploit faulty ciphertexts.*

Assumption 4. *Before every encryption, the state is set to a value that is selected uniformly at random.*

3 Analysis of FSA-Resistance Proof

Our analysis of the FSA-resistance proof escalates as follows. Section 3.1 illustrates that the *attacker model* is ill-defined. Section 3.2 points out harmless physical misassumptions on gate propagation delays. Section 3.3 identifies a fatal error in the reasoning behind the proof. Section 3.4 provides examples of TIs that succumb to FSA.

3.1 Ill-Defined Attacker Model

The attacker model, which underlies the proof, is ill-defined. The biggest problem is that many variations of FSA exist, and it is unclear which variations are covered by the proof. Arribas *et al.* [1, 6] supplemented Assumption 1 with a summary of the original FSA by Li *et al.* [11], but vaguely tag it as an “explanation of the validity of Assumption 1”, so the reader cannot distinguish whether it concerns either an example of a covered FSA or the one and only covered FSA. Two facts strongly suggest it is just an example. First, De Cnudde [6] explicitly acknowledges support for the FSA by Mischke *et al.* [12], which implies that Assumption 1 must be broad enough to cover this type of attack. Second, Arribas *et al.* [1, 6] motivate their work by describing how Moradi *et al.* [13] attack non-glitch-resistant masking schemes, and then suggest that TIs provide a solution. If the FSA by Moradi *et al.* [13] would not be covered in Assumption 1, this whole *background story* would be a facade, *i.e.*, the reader is actively mislead. Experiments performed by Arribas *et al.* [1] simulate the original FSA by Li *et al.* [11], but no conclusions can be drawn from this. By default, experiments in a paper comprise a small subset of the infinitely large set of all possible test cases. Also if numerous FSA variations are covered, it would not be practical to test them all.

The whole situation is evidently confusing, but now it gets truly absurd. The original FSA by Li *et al.* [11], which serves as a supplement to Assumption 1, does not even apply to TIs. This particular attack assumes that the fault sensitivity is constant for a given algorithm input such that repeated evaluations can be used to precisely measure the fault sensitivity. For TIs, however, the fault sensitivity changes with every evaluation due to the random masks. It would thus be nonsense for an attacker to gradually increase the fault intensity until a fault occurs. Note that in their experiments, Arribas *et al.* [1] are only able to apply the attack to TIs because the fault sensitivities are simulated rather than measured. On an actual device, the attack would be impossible.

In our analysis, we try to work around the ambiguities in the most conservative possible way. Despite a plethora of fault injecting techniques, we only consider reductions of the clock period T_{clk}. Note that with heating and under-powering, propagation delays D are not only measured but also increased in possibly complex, non-linear ways. We also exclude spatially local FSA, *e.g.*, the use of a laser for the forced reevaluation of a combinational subcircuit [17], as this might be in conflict with the intended parallelism in Assumption 2. For the actual refutation of the proof, Section 3.3 points out a fatal reasoning error that is generic, *i.e.*, no ties to a specific FSA variation are being made. The counterexamples in Section 3.4 bear a resemblance to FSAs by Li *et al.* [10] and Mischke *et al.* [12]. This evidently complies with the revised proof by De Cnudde [6], which explicitly claims resistance to the FSA by Mischke *et al.* [12]. For the original proof by Arribas *et al.* [1], it becomes fair out of necessity: the only explicitly covered FSA variation [11] is an inapplicable one. In fact, no problem exists given that the counterexamples are optional anyway; Section 3.3 is self-sufficient as proof refutation.

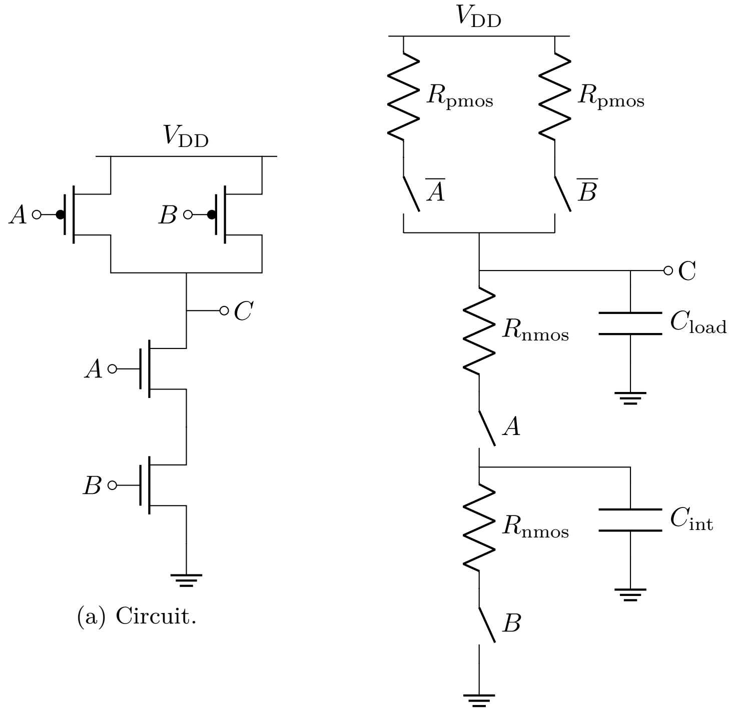
For the revised proof by De Cnudde [6], a few additional ambiguities need to be tackled. Regarding Assumption 3, the author does not comment on undetectable and ineffective faults. An example of the former are identical faults in duplicated hardware. An example of the latter are faulted output shares $(\mathbf{y}_1 \oplus \mathbf{e}_1, \mathbf{y}_2 \oplus \mathbf{e}_2, \mathbf{y}_3 \oplus \mathbf{e}_3)$ where $\mathbf{e}_1 \oplus \mathbf{e}_2 \oplus \mathbf{e}_3 = \mathbf{0}$. Again, to be conservative, we consider every single bit flip as detectable. For Assumption 4, it is unspecified whether the randomly selected value is secret or not. We consider it a secret.

3.2 Deficiencies of The Physical Model

We point out several misassumptions on propagation delays for static *complementary metal–oxide–semiconductor* (CMOS) logic in particular. Section 3.2.1 illustrates the inadequacy of focussing on inter-gate data dependencies and neglecting intra-gate data dependencies. Furthermore, the componentwise delays $D_{\mathsf{G}_1}(X_2, X_3, Y_1)$, $D_{\mathsf{G}_2}(X_1, X_3, Y_2)$, and $D_{\mathsf{G}_3}(X_1, X_2, Y_3)$ used in the FSA-resistance proof are unable to capture the following two physical phenomena: noise and glitches, as discussed in Section 3.2.2 and Section 3.2.3 respectively. As argued in Section 3.2.4, the FSA-resistance proof still stands, given a notational generalization that captures the overlooked phenomena.

3.2.1 Data-Dependent Propagation Delays Are Everywhere

The problem of data-dependent propagation delays is more severe than Li *et al.* [11] and Arribas *et al.* [1] assume. Gate propagation delays cannot accurately be described by a single parameter. Therefore, in addition to *inter*-gate data dependencies, *intra*-gate data dependencies arise. For static CMOS logic, the latter are unforgiving. Consider, for example, a two-input NAND gate, of which the circuit and a *resistor–capacitor* (RC) model are shown in Fig. 3. Simulation results of Rabaey *et al.* [15, Chapter 6], which are repeated here in Table 1, show that all six possible transitions that invert the value of the output C are characterized by distinct propagation delays. These delay differences are significant and thus measurable: most notably, the largest delay is approximately twice as large as the smallest delay. The transition in row four is particularly fast because the load capacitor C_{load} is charged through two parallel paths, having an equivalent resistance $R_{\text{eq}} = R_{\text{nmos}}/2$. The transition in row five is particularly slow because an uncharged internal capacitor C_{int} adds to the load. Note that although inputs A and B are interchangeable on a functional level, this symmetry does not hold on the circuit level: the order of the serialized nMOS transistors matters. For all ten possible transitions where the value of the output C remains unchanged, propagation delay $D = 0$.



(a) Circuit.

(b) RC model.

Figure 3: A two-input NAND gate in static CMOS technology. (a) The circuit consists of two nMOS and two pMOS transistors. (b) The load capacitance C_{load} comprises an aggregate of all gates driven by the NAND gate.

Evidently, the common distinction between insecure AND/NAND/OR/NOR gates and secure XOR/XNOR gates [11, 1, 9] is over-simplistic. All gates, including NOT gates, have data-dependent propagation delays. Note also that the data dependencies given in Eq. (1) are semi-accurate at best: if the output of the AND/OR gate remains unchanged, $D = 0$ irrespective of the value of input A. Fig. 4 includes an example of $D = 0$, *i.e.*, the first out of three output nodes.

3.2.2 Noise and Time-Variant Fault Sensitivity

Unfortunately, electronic circuits are subject to noise [15], *i.e.*, irreproducible deviations from the nominal behavior caused by randomly moving particles. For example, the resistive

Table 1: Data-dependent propagation delays of the two-input NAND gate shown in Fig. 3, as simulated by Rabaey *et al.* [15, Chapter 6] for CMOS transistors having a channel length of 0.25 µm. Although this technology is now obsolete, the delay differences originate from unavoidable circuit asymmetries and, therefore, still exist today in similar proportions.

Input A	Input B	Output C	Delay D
$0 \rightarrow 1$	$0 \rightarrow 1$	$1 \rightarrow 0$	69 ps
1	$0 \rightarrow 1$	$1 \rightarrow 0$	62 ps
$0 \rightarrow 1$	1	$1 \rightarrow 0$	50 ps
$1 \rightarrow 0$	$1 \rightarrow 0$	$0 \rightarrow 1$	35 ps
1	$1 \rightarrow 0$	$0 \rightarrow 1$	76 ps
$1 \rightarrow 0$	1	$0 \rightarrow 1$	57 ps

elements in Fig. 3b exhibit *Johnson–Nyquist noise*, which is the thermal agitation of *charge carriers*. Hence, electrical signals such as currents and voltages as a function of time are not deterministic but stochastic in nature. On gate level, these noisy signals manifest as the following two phenomena, both of which result in a time-variant fault sensitivity. First, for a combinatorial circuit that responds to a given input transition, propagation delays D are more accurately described by a Gaussian-like distribution than by constant. Second, if the setup and/or hold times of a flip-flop are violated, it enters a *metastable* state that eventually resolves to either 0 or 1 depending on internal noise sources. In conclusion, it is a misassumption of Arribas *et al.* [1, 6] that a clear-cut threshold D_{G_i} separating correct and faulty outputs exist.

3.2.3 Inability to Model Glitches

Another overlooked phenomenon is that in the presence of glitches, propagation delays are hard to define using a single variable D. Although for the second, non-glitching output node in Fig. 4, D obviously relates to the *falling edge*, for the third output node, there is one rising and one falling edge to consider. An argument for $D = 0$ can be made: when the clock period is reduced to $T'_{clk} \approx 0$, the register samples and also ends up storing the correct value if the rising edge comes after the *hold time* has passed. Another justifiable choice is to relate D to second, falling edge. Conceivably, if a simple glitch comprising two signal transitions already sparks debate, the situation becomes increasingly unsustainable for three or more edges within a single clock period. To capture the effects of noise and glitches alike, we suggest replacing delays D by a function $\mathbb{P}_{\text{correct}}(T'_{clk})$, *i.e.*, the probability that a flip-flop ends up storing the correct value for a given reduced clock period $T'_{clk} \in [0, T_{clk}]$. Note that the sloped edges of $\mathbb{P}_{\text{correct}}(T'_{clk})$ in Fig. 4 represent noise. The proposed function directly embodies what FSA exploits: information on the correctness of the registered output for a given fault intensity. Similar to propagation delays, such a function can also be defined for a set of flip-flops rather than a single flip-flop only.

3.2.4 Consequences

For the FSA-resistance proof, there is no fundamental problem yet. Nevertheless, the componentwise delays should be redefined such that phenomena occurring in static CMOS logic can be adequately modeled. In absence of glitches, one could resort to Eq. (4), where superscripts (1) and (2) refer to previous and new input values respectively. This definition slightly differs from Arribas *et al.* [1] in order to solve the following problem: as component functions G_i are usually non-injective, distinct reset values $(\boldsymbol{\chi}_2, \boldsymbol{\chi}_3)$ can map to the same $Y_1 \triangleq \mathsf{G}_1(\boldsymbol{\chi}_2, \boldsymbol{\chi}_3)$, yet result in different charges on the internal circuit nodes and thus different propagation delays, and similarly for G_2 and G_3. Note that the componentwise delay functions of De Cnudde [6] are discarded by default due to their

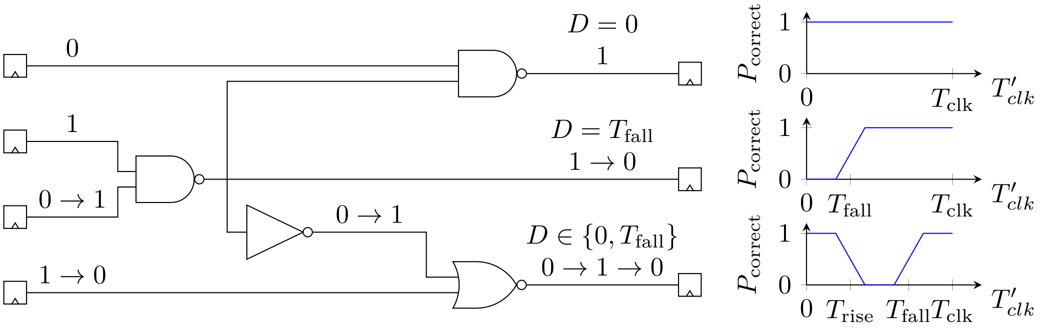


Figure 4: In the presence of noise, there is gray zone between correct and faulty outputs. In the presence of glitches, propagation delay D is hard to define.

imprecise specification. Despite the improvement, Eq. (4) maintains the irony of not being able to describe glitches even though TIs are specifically advertised as a glitch-resistant masking scheme. Equation (5) solves this problem, and has ability to model noise as well.

$$\begin{aligned} &D_{\mathsf{G}_1}(X_2^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}),\\ &D_{\mathsf{G}_2}(X_1^{(1)}, X_3^{(1)}, X_1^{(2)}, X_3^{(2)}),\\ &D_{\mathsf{G}_3}(X_1^{(1)}, X_2^{(1)}, X_1^{(2)}, X_2^{(2)}). \end{aligned} \tag{4}$$

$$\begin{aligned} &P_{\text{correct},\mathsf{G}_1}(X_2^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}, T'_{clk}),\\ &P_{\text{correct},\mathsf{G}_2}(X_1^{(1)}, X_3^{(1)}, X_1^{(2)}, X_3^{(2)}, T'_{clk}),\\ &P_{\text{correct},\mathsf{G}_3}(X_1^{(1)}, X_2^{(1)}, X_1^{(2)}, X_2^{(2)}, T'_{clk}). \end{aligned} \tag{5}$$

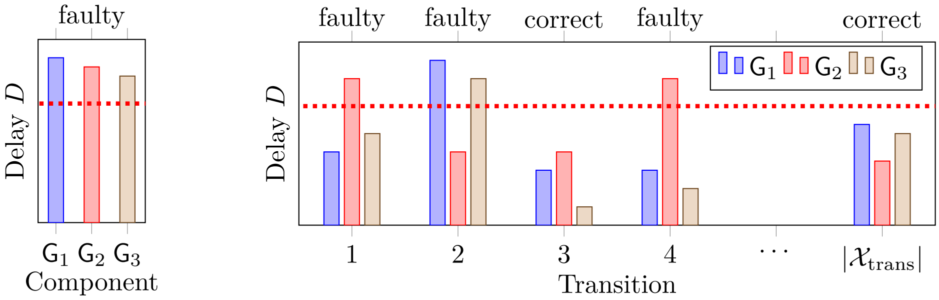
3.3 The Incompleteness Fallacy

A peculiarity of the FSA-resistance proof is that the made assumptions are not explicitly incorporated. Most notably, Assumption 4 stipulates that the initial state value should be selected uniformly at random, but this particular probability distribution never comes back in the actual proof, *e.g.*, through a formal derivation making use of probability theory. Not surprisingly for a proof in which *stepping stones* are missing, a fatal flaw arises.

The backbone of the proof is that the correctness of the output (Y_1, Y_2, Y_3) supposedly only depends on one component function G_i, thereby preserving the secrecy of the input X through the incompleteness property. This reasoning is wrong: the maximum propagation delay $D_{(\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3)} \triangleq \max(D_{\mathsf{G}_1}, D_{\mathsf{G}_2}, D_{\mathsf{G}_3})$, which can be measured by the attacker, depends on all three input shares and thus also reveals information on all three input shares. For componentwise delays as defined in Eq. (4), this exploit is formalized as follows. Consider the set $\mathcal{X}_{\text{trans}}$ of all possible shared input transitions. For Arribas *et al.* [1], the cardinality $|\mathcal{X}_{\text{trans}}| = 2^{\sigma\,\lambda}$; for De Cnudde [6], $|\mathcal{X}_{\text{trans}}| = 4^{\sigma\,\lambda}$. For any given fault intensity D', the set $\mathcal{X}_{\text{trans}}$ can be partitioned into the two subsets that are defined in Eq. (6). For each evaluation, the attacker knows in which of the two sets the actual transition resides. Hence, unless the condition in Eq. (7) is true, which is unlikely to be the case in practice, the TI is vulnerable to FSA. Note that if Eq. (7) is true, a similar condition for $\mathcal{X}_{\text{trans,faulty}}(D')$ is also true. Figure 5 illustrates the fallacy; the fault intensity D' is represented by the red dotted line.

$$\begin{aligned} \mathcal{X}_{\text{trans,correct}}(D') \triangleq \big\{\big(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}, X_1^{(2)}, X_2^{(2)}, X_3^{(2)}\big) \in \mathcal{X}_{\text{trans}} \mid \\ \big(D_{\mathsf{G}_1}\big(X_2^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}\big) < D'\big) \wedge \big(D_{\mathsf{G}_2}\big(X_1^{(1)}, X_3^{(1)}, X_1^{(2)}, X_3^{(2)}\big) < D'\big) \wedge \\ \big(D_{\mathsf{G}_3}\big(X_1^{(1)}, X_2^{(1)}, X_1^{(2)}, X_2^{(2)}\big) < D'\big)\big\}, \\ \mathcal{X}_{\text{trans,faulty}}(D') \triangleq \mathcal{X}_{\text{trans}} \setminus \mathcal{X}_{\text{trans,correct}}(D'). \end{aligned} \tag{6}$$

$$\begin{gathered} \forall D' \in [0, T_{\text{clk}}], \forall X \in \{0,1\}^\lambda, \\ \big|\big\{\big(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}, X_1^{(2)}, X_2^{(2)}, X_3^{(2)}\big) \in \mathcal{X}_{\text{trans,correct}}(D') \mid \\ X_1^{(2)} \oplus X_2^{(2)} \oplus X_3^{(2)} = X\big\}\big| = \big|\mathcal{X}_{\text{trans,correct}}(D')\big|/2^\lambda. \end{gathered} \tag{7}$$



(a) 2D view.

(b) 3D view.

Figure 5: The longest propagation delay D in each component function G_i. (a) The static, two-dimensional view of Arribas *et al.* [1, 6]. (b) A more dynamic, three-dimensional view.

3.4 Counterexamples

Our refutation of the proof is now solidified by providing a counterexample both for the original work by Arribas *et al.* [1] and for the hardened version by De Cnudde [6], in which Assumption 4 respectively does not and does exist. More precisely, for a TI of order $\delta = 1$, we specify an instance of the componentwise delays D_{G_i} in Eq. (4) that succumbs to FSA. As noise and glitches cannot adequately be captured by Eq. (4), we make abstraction of these phenomena.

3.4.1 Excluding Assumption 4

Consider an arbitrary invertible, quadratic function $\mathsf{G} : \{0,1\}^\lambda \to \{0,1\}^\lambda$, which can be thought of as an S-box. For its arbitrary TI, let the input shares initially be zero, which is a typical reset value for registers. As specified in Eq. (8), let the componentwise delays D_{G_1}, D_{G_2}, and D_{G_3} be zero if the respective function outputs remain unchanged. In practice, this condition is true if the λ output nodes of each G_i are free of glitches. If the output of a G_i changes, let the delay be equal to an arbitrary strictly positive constant, of which the *unit of measurement* is omitted. In practice, this condition is approximately true for TIs where each G_i is realized as a *lookup table* (LUT). Such realizations require a total of $3 \cdot 4^\lambda\, \lambda$ hardwired bits and are thus area-inefficient for typical values of λ, *e.g.*, $\lambda \in \{4, 8\}$, but we are free to adopt any unconventional piece of hardware that complies with the terms of the proof.

$$
\begin{aligned}
D_{\mathsf{G}_1} &\triangleq \begin{cases} 0 & \text{, if } \mathsf{G}_1\big(X_2^{(2)}, X_3^{(2)}\big) = \mathsf{G}_1(\mathbf{0},\mathbf{0}) \\ 3 & \text{, otherwise,} \end{cases} \\
D_{\mathsf{G}_2} &\triangleq \begin{cases} 0 & \text{, if } \mathsf{G}_2\big(X_1^{(2)}, X_3^{(2)}\big) = \mathsf{G}_2(\mathbf{0},\mathbf{0}) \\ 3 & \text{, otherwise,} \end{cases} \qquad \text{where } X_1^{(1)} \triangleq X_2^{(1)} \triangleq X_3^{(1)} \triangleq \mathbf{0}. \\
D_{\mathsf{G}_3} &\triangleq \begin{cases} 0 & \text{, if } \mathsf{G}_3\big(X_1^{(2)}, X_2^{(2)}\big) = \mathsf{G}_3(\mathbf{0},\mathbf{0}) \\ 3 & \text{, otherwise,} \end{cases}
\end{aligned} \tag{8}
$$

Due to the independency of component functions, $D_{(\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3)} \triangleq \max(D_{\mathsf{G}_1}, D_{\mathsf{G}_2}, D_{\mathsf{G}_3})$. Hence, we obtain Eq. (9) from Eq. (8).

$$
\begin{aligned}
D_{(\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3)} &= \begin{cases} 0 & \text{, if } \big(X_1^{(2)}, X_2^{(2)}, X_3^{(2)}\big) \in \mathcal{D}_0 \\ 3 & \text{, otherwise,} \end{cases} \\
\text{where } \mathcal{D}_0 = \big\{(X_1, X_2, X_3) \mid & \big(\mathsf{G}_1(X_2, X_3) = \mathsf{G}_1(\mathbf{0},\mathbf{0})\big) \\
\wedge \big(\mathsf{G}_2(X_1, X_3) = \mathsf{G}_2(\mathbf{0},\mathbf{0})\big) &\wedge \big(\mathsf{G}_3(X_1, X_2) = \mathsf{G}_3(\mathbf{0},\mathbf{0})\big)\big\}
\end{aligned} \tag{9}
$$

A necessary but insufficient condition for $D_{(\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3)} = 0$ is given in Eq. (10). It follows that $|\mathcal{D}_0| \leq 2^{\lambda(\sigma-1)}$. As $(\mathbf{0},\mathbf{0},\mathbf{0}) \in \mathcal{D}_0$, it also holds that $|\mathcal{D}_0| \geq 1$. The exact value of $|\mathcal{D}_0| \in [1, 2^{\lambda(\sigma-1)}]$ can easily be computed for a given TI.

$$
\begin{aligned}
& \big(X_1^{(2)}, X_2^{(2)}, X_3^{(2)}\big) \in \mathcal{D}_0 \\
\implies & \mathsf{G}_1\big(X_2^{(2)}, X_3^{(2)}\big) \oplus \mathsf{G}_2\big(X_1^{(2)}, X_3^{(2)}\big) \oplus \mathsf{G}_3\big(X_1^{(2)}, X_2^{(2)}\big) \\
& = \mathsf{G}_1(\mathbf{0},\mathbf{0}) \oplus \mathsf{G}_2(\mathbf{0},\mathbf{0}) \oplus \mathsf{G}_3(\mathbf{0},\mathbf{0}) \\
& \implies \mathsf{G}\big(X^{(2)}\big) = \mathsf{G}(\mathbf{0}) \implies X^{(2)} = \mathbf{0}
\end{aligned} \tag{10}
$$

If the attacker reduces the clock period such that $D_{(\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3)} = 2$ is the threshold between a correct and a faulty computation, the data-dependent statistic in Eq. (11) arises. A correct output implies that $X^{(2)} = \mathbf{0}$, *i.e.*, a secret value is revealed in its entirety. Note also that the specification of the componentwise delays in Eq. (8) can be relaxed to accommodate a more realistic attack. For the '*otherwise*' cases in Eq. (8), D_{G_i} is not necessarily constant and may depend on noise, process variations, and both shares of $X^{(2)}$ as long as all D_{G_i}'s exceed a predefined threshold. Observe that the presented attack is similar to the aforementioned FSA variation where identical inputs in subsequent clock cycles are spotted [10, 12]; the difference lies in the additional complexity of component functions G_i being non-injective.

$$
\mathbb{P}\big(D_{(\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3)} < 2\big) = \begin{cases} |\mathcal{D}_0|/2^{\lambda\,(\sigma-1)} & \text{, if } X^{(2)} = \mathbf{0} \\ 0 & \text{, otherwise.} \end{cases} \tag{11}
$$

3.4.2 Including Assumption 4

Consider an arbitrary invertible, affine function $\mathsf{G} : \{0,1\}^\lambda \to \{0,1\}^\lambda$. Its TI operates on $\sigma_{\text{in}} = \sigma_{\text{out}} = 2$ shares and is constructed as follows: $\mathsf{G}_1(\mathbf{x}_1) \triangleq \mathsf{G}(\mathbf{x}_1)$ and $\mathsf{G}_2(\mathbf{x}_2) \triangleq \mathsf{G}(\mathbf{x}_2)$. The componentwise delays D_{G_i} are given in Eq. (12), where the initial state is drawn uniformly at random for each evaluation. As G_1 and G_2 are identical, we assume that D_{G_1} and D_{G_2} are identical as well. As $\mathsf{G} \triangleq \mathsf{G}_1 \triangleq \mathsf{G}_2$ is injective, it is reasonable to assume that

$D_{\mathsf{G}_i} = 0$ if and only if input share X_i remains unchanged. Furthermore, for one particular nonvoid transition, the strictly positive delay is particularly small.

$$D_{\mathsf{G}_1} \triangleq \begin{cases} 0 & \text{, if } X_1^{(1)} = X_1^{(2)} \\ 1 & \text{, if } X_1^{(1)} = \mathbf{1} \text{ and } X_1^{(2)} = \mathbf{0} \\ 3 & \text{, otherwise,} \end{cases}$$
$$D_{\mathsf{G}_2} \triangleq \begin{cases} 0 & \text{, if } X_2^{(1)} = X_2^{(2)} \\ 1 & \text{, if } X_2^{(1)} = \mathbf{1} \text{ and } X_2^{(2)} = \mathbf{0} \\ 3 & \text{, otherwise.} \end{cases} \tag{12}$$

Due to the independency of component functions, $D_{(\mathsf{G}_1,\mathsf{G}_2)} \triangleq \max(D_{\mathsf{G}_1}, D_{\mathsf{G}_2})$. Again, the attacker reduces the clock period such that $D_{(\mathsf{G}_1,\mathsf{G}_2)} = 2$ is the threshold between a correct and a faulty computation. Through the *tree diagram* in Fig. 6, we obtain the compromising statistic in Eq. (13), *i.e.*, the probability that the output is correct is slightly higher if $X^{(2)} = \mathbf{0}$. To independently verify the correctness of Eq. (13), for all $\lambda \in [1, 6]$, we let a MATLAB script evaluate Eq. (12) and $D_{(\mathsf{G}_1,\mathsf{G}_2)}$ for all $4^{\sigma\,\lambda}$ possible input transitions.

$$\mathbb{P}\big(D_{(\mathsf{G}_1,\mathsf{G}_2)} < 2\big) = \begin{cases} 2^{-\lambda\,\sigma}\big(1 + 3 \cdot 2^{-\lambda\,(\sigma-1)}\big) & \text{, if } X^{(2)} = \mathbf{0} \\ 2^{-\lambda\,\sigma}\big(1 + 2 \cdot 2^{-\lambda\,(\sigma-1)}\big) & \text{, otherwise.} \end{cases} \tag{13}$$

4 Concluding Remarks

The FSA-resistance proof [1, 6] also applies to glitch-resistant masking schemes other than TIs, such as the scheme of Roche and Prouff [16]. The incompleteness fallacy described in Section 3.3 refutes those instantiations of the proof in a similar way, so we presume it is merely a formality to devise counterexamples, as we did for TIs in Section 3.4. Another remark is that the intra-gate data-dependent delays described in Section 3.2.1 fundamentally harm FSA countermeasures that rely on the balancing of gate depth levels [11, 9, 8].

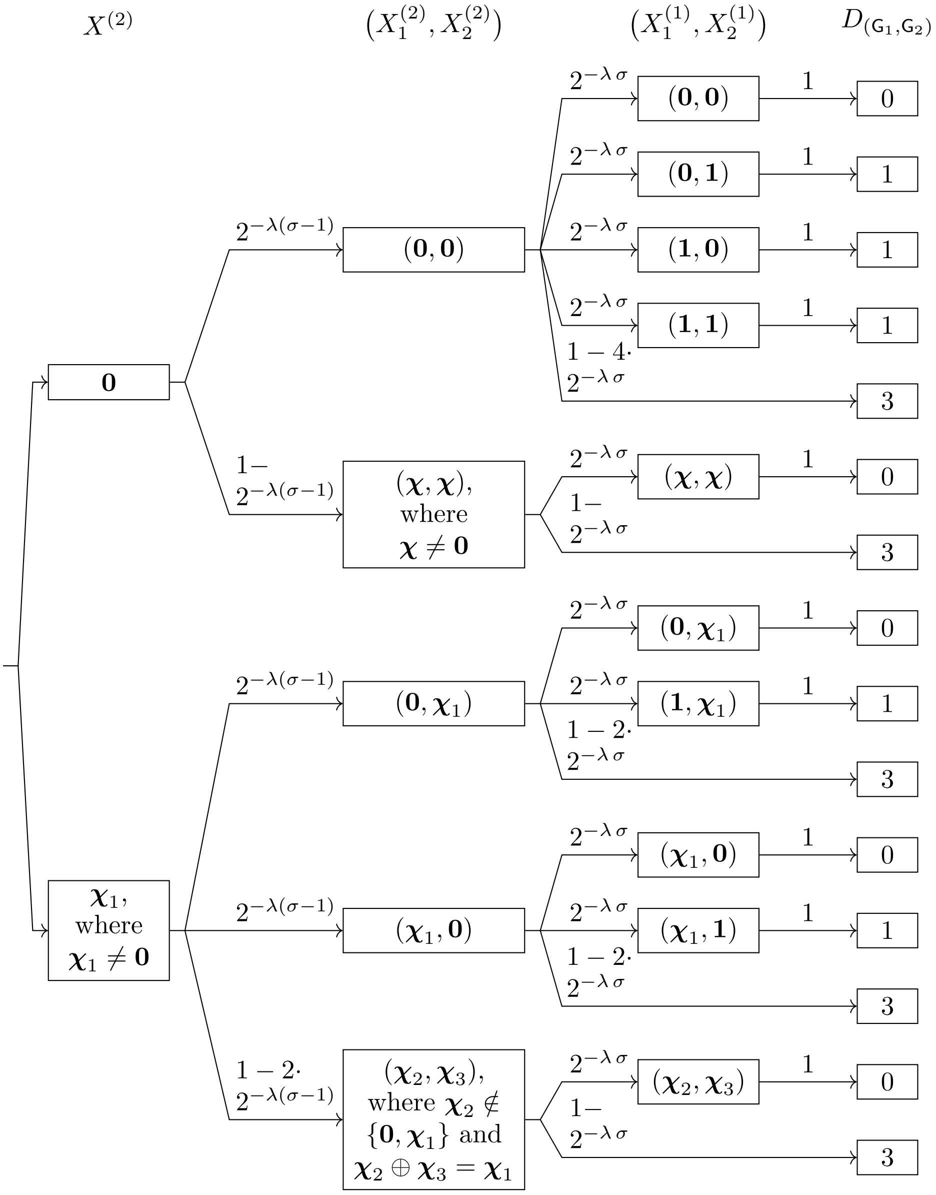

$X^{(2)}$
$(X_1^{(2)}, X_2^{(2)})$
$(X_1^{(1)}, X_2^{(1)})$
$D_{(\mathsf{G}_1,\mathsf{G}_2)}$
$\mathbf{0}$
$2^{-\lambda(\sigma-1)}$
$(\mathbf{0},\mathbf{0})$
$2^{-\lambda\sigma}$
$(\mathbf{0},\mathbf{0})$
1
0
$2^{-\lambda\sigma}$
$(\mathbf{0},\mathbf{1})$
1
1
$2^{-\lambda\sigma}$
$(\mathbf{1},\mathbf{0})$
1
1
$2^{-\lambda\sigma}$
$(\mathbf{1},\mathbf{1})$
1
1
$1-4\cdot 2^{-\lambda\sigma}$
3
$1-2^{-\lambda(\sigma-1)}$
$(\boldsymbol{\chi},\boldsymbol{\chi})$, where $\boldsymbol{\chi} \neq \mathbf{0}$
$2^{-\lambda\sigma}$
$(\boldsymbol{\chi},\boldsymbol{\chi})$
1
0
$1-2^{-\lambda\sigma}$
3
$\boldsymbol{\chi}_1$, where $\boldsymbol{\chi}_1 \neq \mathbf{0}$
$2^{-\lambda(\sigma-1)}$
$(\mathbf{0},\boldsymbol{\chi}_1)$
$2^{-\lambda\sigma}$
$(\mathbf{0},\boldsymbol{\chi}_1)$
1
0
$2^{-\lambda\sigma}$
$(\mathbf{1},\boldsymbol{\chi}_1)$
1
1
$1-2\cdot 2^{-\lambda\sigma}$
3
$2^{-\lambda(\sigma-1)}$
$(\boldsymbol{\chi}_1,\mathbf{0})$
$2^{-\lambda\sigma}$
$(\boldsymbol{\chi}_1,\mathbf{0})$
1
0
$2^{-\lambda\sigma}$
$(\boldsymbol{\chi}_1,\mathbf{1})$
1
1
$1-2\cdot 2^{-\lambda\sigma}$
3
$1-2\cdot 2^{-\lambda(\sigma-1)}$
$(\boldsymbol{\chi}_2,\boldsymbol{\chi}_3)$, where $\boldsymbol{\chi}_2 \notin \{\mathbf{0},\boldsymbol{\chi}_1\}$ and $\boldsymbol{\chi}_2 \oplus \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1$
$2^{-\lambda\sigma}$
$(\boldsymbol{\chi}_2,\boldsymbol{\chi}_3)$
1
0
$1-2^{-\lambda\sigma}$
3

Figure 6: Tree diagram representing the distribution of $D_{(\mathsf{G}_1,\mathsf{G}_2)} \triangleq \max(D_{\mathsf{G}_1}, D_{\mathsf{G}_2})$, where D_{G_1} and D_{G_2} are defined in Eq. (12). Note that $\sigma = 2$.

References

[1] Victor Arribas, Thomas De Cnudde, and Danilo Šijačić. Glitch-resistant masking schemes as countermeasure against fault sensitivity analysis. In *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2018)*, pages 27–34. IEEE, September 2018.

[2] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, August 1997.

[3] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Higher-order threshold implementations. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security*, volume 8874 of *Lecture Notes in Computer Science*, pages 326–343. Springer, December 2014.

[4] Thomas De Cnudde, Maik Ender, and Amir Moradi. Hardware masking, revisited. *IACR Transactions on Cryptographic Hardware and Embedded Systems (CHES 2018)*, 2018(2):123–148, 2018.

[5] Joan Daemen. Changing of the guards: A simple and efficient method for achieving uniformity in threshold sharing. In Wieland Fischer and Naofumi Homma, editors, *19th Conference on Cryptographic Hardware and Embedded Systems (CHES 2017)*, volume 10529 of *Lecture Notes in Computer Science*, pages 137–153. Springer, September 2017.

[6] Thomas De Cnudde. *Cryptography Secured Against Side-Channel Attacks.* PhD thesis, KU Leuven, 2018.

[7] Christoph Dobraunig, Maria Eichlseder, Thomas Korak, Stefan Mangard, Florian Mendel, and Robert Primas. SIFA: Exploiting ineffective fault inductions on symmetric cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(3):547–572, 2018.

[8] Hassan Eldib, Meng Wu, and Chao Wang. Synthesis of fault-attack countermeasures for cryptographic circuits. In Swarat Chaudhuri and Azadeh Farzan, editors, *28th International Conference on Computer Aided Verification (CAV 2016)*, volume 9780 of *Lecture Notes in Computer Science*, pages 343–363. Springer, July 2016.

[9] Nahid Farhady Ghalaty, Aydin Aysu, and Patrick Schaumont. Analyzing and eliminating the causes of fault sensitivity analysis. In Gerhard P. Fettweis and Wolfgang Nebel, editors, *Design, Automation & Test in Europe Conference & Exhibition (DATE 2014)*, pages 1–6. IEEE, March 2014.

[10] Yang Li, Kazuo Ohta, and Kazuo Sakiyama. An extension of fault sensitivity analysis based on clockwise collision. In Miroslaw Kutylowski and Moti Yung, editors, *8th Conference on Information Security and Cryptology (Inscrypt 2012)*, volume 7763 of *Lecture Notes in Computer Science*, pages 46–59. Springer, November 2012.

[11] Yang Li, Kazuo Ohta, and Kazuo Sakiyama. New fault-based side-channel attack using fault sensitivity. *IEEE Transactions on Information Forensics and Security*, 7(1):88–97, February 2012.

[12] Oliver Mischke, Amir Moradi, and Tim Güneysu. Fault sensitivity analysis meets zero-value attack. In Assia Tria and Dooho Choi, editors, *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2014)*, pages 59–67. IEEE, September 2014.

[13] Amir Moradi, Oliver Mischke, Christof Paar, Yang Li, Kazuo Ohta, and Kazuo Sakiyama. On the power of fault sensitivity analysis and collision side-channel attacks in a combined setting. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 292–311. Springer, September 2011.

[14] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *Journal of Cryptology*, 24(2):292–321, 2011.

[15] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. *Digital Integrated Circuits.* Pearson, second edition, 2003.

[16] Thomas Roche and Emmanuel Prouff. Higher-order glitch free implementation of the AES using secure multi-party computation protocols. *Journal of Cryptographic Engineering*, 2(2):111–127, June 2012.

[17] Falk Schellenberg, Markus Finkeldey, Nils Gerhardt, Martin Hofmann, Amir Moradi, and Christof Paar. Large laser spots and fault sensitivity analysis. In William H. Robinson, Swarup Bhunia, and Ryan Kastner, editors, *Symposium on Hardware Oriented Security and Trust (HOST 2016)*, pages 203–208. IEEE, May 2016.