Threshold Implementations Are Not Provably Secure Against Fault Sensitivity Analysis

Jeroen Delvaux[0000−0003−0684−8427]

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 Sijačić 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 is incorrect.

Keywords: Threshold implementations · Fault sensitivity analysis · Provably secure

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 such 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 (TIs) [4,17] are a popular masking method as few assumptions about the underlying hardware are made in their security proofs.

Unfortunately, an attacker 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 algorithm outputs are exploitable through, e.g., a differential fault analysis (DFA) [3], ciphers 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. Sadly, several types of fault attacks, including a statistical ineffective fault attack (SIFA) [8] and a fault sensitivity analysis

1 The latest version of this article appears in the Cryptology ePrint Archive (https://eprint.iacr.org/2020/400). If you downloaded this file elsewhere, it is an unauthorized reupload.
(FSA) do not necessarily require faulty outputs; correct outputs and/or knowledge of whether the outputs are correct or faulty might suffice. Ironically, redundant implementations with output suppression conveniently provide this one bit of correctness information to the attacker. In an attempt to *kill two birds with one stone* Arribas et al. prove that TIs, which were previously known to only resist side-channel analysis, also resist FSA. One coauthor, De Cnudde, later imposed additional conditions for the proof to hold, thereby significantly increasing a TI's intake of random bits.

### 1.1 Contribution

We argue that the FSA-resistance proof, both in its original form by Arribas et al. and in its revised form by De Cnudde, contains a fatal mathematical error. To strengthen our claim, we specify instances of TIs that succumb to FSA. We also point out that both versions of the proof are riddled with questionable abstractions of physical phenomena that occur in static complementary metal–oxide–semiconductor (CMOS) logic, *i.e.*, several abstractions that are acceptable for the original side-channel-resistance proof cause anomalies in the case of FSA.

### 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. Section 2.2 and Section 2.3 introduce the fundamentals of FSA and TIs respectively. Section 2.4 recapitulates the FSA-resistance proof.

#### 2.1 Notation

Variables and constants are denoted by characters from the Latin and Greek alphabets 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 $\lambda$-bit vectors is denoted by $\{0, 1\}^\lambda$. Functions are printed in a sans-serif font, *e.g.*, $G$.

---

2 To be interpret in the idiom’s symbolic sense. We do not condone cruelty to animals.
2.2 Fault Sensitivity Analysis

The propagation delay $D$ of a function $G$ implemented as combinational logic depends on the value of its input data. Li et al. [14] 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 constant $\delta$. 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 $D = \delta_{\text{NOT}} + \delta_{\text{XOR}}$.

\[
D = \begin{cases} 
\delta_{\text{AND}}, & \text{if } A = 0 \\
\delta_{\text{NOT}} + \delta_{\text{AND}}, & \text{otherwise}
\end{cases}, \quad D = \begin{cases} 
\delta_{\text{OR}}, & \text{if } A = 1 \\
\delta_{\text{NOT}} + \delta_{\text{OR}}, & \text{otherwise}
\end{cases}
\]  

(1)

Also for larger circuits, such as a substitution box (S-box) $G : X \rightarrow Y$ of a symmetric-key cipher where the input $X$ and the output $Y$ are stored in registers, the propagation delay $D$ for settling $Y$ depends on $X$. Consequentially, the register storing $Y$ has a sensitivity to setup-time violations that depends on $X$. In the original FSA by Li et al. [14], this sensitivity is measured by fixing $X$ to a constant value and repeatedly evaluating $G$ such that the fault intensity, i.e., the intensity level of a fault injection tool, is gradually increased until $Y$ becomes erroneous. For example, an attacker can progressively decrease the time difference $T_{\text{clk}}$ between two consecutive rising edges of the clock signal. Alternatively, the clock signal is unmodified, but propagation delays $D$ are progressively increased, either by increasing the temperature or by decreasing the supply voltage [19]. Faulty outputs $Y$ 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 in the last round of an encryption function that is subjected to a global fault injection method such as under-powering, the correctness of individual S-box outputs $Y$ can be assessed rather than the correctness of the complete ciphertext.

In its original form by Li et al. [14], FSA requires a mathematical model of the data-dependent fault sensitivity, i.e., knowledge of the circuit or even the layout is required. Several variations of FSA, which we subdivide into two
categories, avoid this burden. The first category of variations as initiated by Li et al. [12] exploits that 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. Similarly, Mischke et al. [15] exploit that for certain S-box implementations, the pair \((X, Y) = (0, 0)\) evaluates with an exceptionally low propagation delay.

For a second category of variations proposed by Li et al. [13] and Moradi et al. [16], 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.

FSA is not to be confused with differential fault intensity analysis (DFIA) [11]. Both techniques require changes of the fault intensity, but the interval of interest differs: FSA exploits the boundary between correct and faulty outputs \(Y\), whereas DFIA exploits boundaries between faulty outputs \(Y\) having \(1, 2, 3, \cdots\) erroneous bits. Another, implied difference is that DFIA requires knowledge of \(Y\), whereas FSA might not.

2.3 Threshold Implementations

In additive Boolean masking schemes, secrets \(x \in \{0, 1\}^\lambda\) are randomly and uniformly split into \(\sigma\) shares according to Definition 1, thereby achieving the provable property given in Lemma 1 [17]. One way to meet Definition 1 is to first select \((\sigma - 1)\) masks \(m\) randomly, uniformly, and independently from \(\{0, 1\}^\lambda\), followed by the computation in Eq. (3).

**Definition 1 (Uniformity).** A secret \(x \in \{0, 1\}^\lambda\) is randomly and uniformly split into \(\sigma\) shares, i.e., \(x_1, x_2, \cdots, x_\sigma \in \{0, 1\}^\lambda\), if and only if the probability mass function (PMF) of \((X_1, X_2, \cdots, X_\sigma)\) given \(X = x\) is given in (2).

\[
P((X_1, X_2, \cdots, X_\sigma) = (x_1, x_2, \cdots, x_\sigma) \mid X = x) =
\begin{cases}
    2^{-\lambda(\sigma-1)}, & \text{if } x_1 \oplus x_2 \oplus \cdots \oplus x_\sigma = x \\
    0, & \text{otherwise}.
\end{cases}
\]

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

\[
x_1 = m_1, x_2 = m_2, \cdots, x_{\sigma-1} = m_{\sigma-1}, x_\sigma = x \oplus m_1 \oplus m_2 \oplus \cdots \oplus m_{\sigma-1}. \tag{3}
\]

For a function of the form \(G : \{0, 1\}^\lambda \to \{0, 1\}^\eta\), a TI of \(G\) transforms \(\sigma_{\text{in}}\) shares of \(G\)’s input \(x\) into \(\sigma_{\text{out}}\) shares of \(G\)’s output \(y \equiv G(x)\), and consists of \(\sigma_{\text{out}}\) component functions \(G_i : \{0, 1\}^\lambda \times \{0, 1\}^\lambda \times \cdots \times \{0, 1\}^\lambda \to \{0, 1\}^\eta\) such that the correctness and \(\gamma\)-th order incompleteness requirements in Definition 2 and Definition 3 respectively are met. A TI where \(\gamma = 1\) and \(\sigma_{\text{in}} = \sigma_{\text{out}} = 3\), which
Arribas et al. [27] use as an example to develop their FSA-resistance proof, may involve computations $y_1 \triangleq G_1(x_2, x_3)$, $y_2 \triangleq G_2(x_1, x_3)$, and $y_3 \triangleq G_3(x_1, x_2)$ 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., $\tau = 1$, TIs are trivially constructed by setting $\sigma_{in} = \sigma_{out} = 2$ and letting $G_1(x_1) \triangleq G(x_1)$ and $G_2(x_2) \triangleq G(x_2)$.

**Definition 2 (Correctness).** The list of component functions, i.e., $G_1, G_2, \cdots, G_{\sigma_{out}}$, is correct if and only if it holds for all tuples of shares $(x_1, x_2, \cdots, x_{\sigma_{in}}) \in \{0, 1\}^\lambda \times \{0, 1\}^\lambda \times \cdots \times \{0, 1\}^\lambda$ that $G_1(x_1, x_2, \cdots, x_{\sigma_{in}}) \oplus G_2(x_1, x_2, \cdots, x_{\sigma_{in}}) \oplus \cdots \oplus G_{\sigma_{out}}(x_1, x_2, \cdots, x_{\sigma_{in}}) = G(x_1 \oplus x_2 \oplus \cdots \oplus x_{\sigma_{in}}) = G(x)$.

**Definition 3 (Incompleteness).** The list of component functions, i.e., $G_1, G_2, \cdots, G_{\sigma_{out}}$, is incomplete to the $\gamma$th order if and only if any out of $(\sigma_{out})$ combinations of component functions $G_i$ does not depend on at least one input share $X_i$.

![Fig. 2: A first-order TI.](image)

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

To resist side-channel attacks of a certain nature, e.g., electromagnetic emissions, it is crucial that the corresponding physical leakages $L_G$ of each component function $G_i$ independently contribute to the total leakage $L$. For a TI where the $G_i$’s are evaluated in parallel, as previously shown in Fig. 2, the total leakage $L$ observed by the attacker then becomes linear in the componentwise leakages $L_G$, thereby allowing Theorem 2 to be proven [4,17]. In practice, the assumption of independent leakages might only be approximately correct [5]: 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 [19], 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 a TI with parallel components). For a TI having order \( \gamma \in \mathbb{N}_0 \) and operating on a secret \( X \) that is randomly and uniformly split into \( \sigma_{\text{in}} \) shares, it holds for any physically leaked variable of the form \( L = L_{G_1} + L_{G_2} + \cdots + L_{G_{\sigma_{\text{out}}}} \) that the \( \gamma^{\text{th}} \) statistical moment of \( L \) is independent of \( X \).

For a composition of two functions, \( G \circ F \), TIs of \( G \) and \( F \) cannot simply be put in series. First, a register layer should separate both TIs \cite{17,18} 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 \( F_i \) \cite{17,18} or through a form of remasking \cite{6}. Such measures are indispensable for block ciphers, which can be understood as a composition of identical round functions, \( i.e., G \circ G \circ \ldots \circ G \).

2.4 FSA-Resistance Proof

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

Original Version Despite a demonstration by Moradi et al. \cite{16} that several masking schemes are vulnerable to FSA, Arribas et al. \cite{2} argue that TIs are provably secure thanks to their incompleteness property\(^3\). 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. \cite{14} where the fault intensity is gradually increased until an erroneous output appears.

Assumption 1 FSA relies on the measurement of propagation delays.

Assumption 2 The component functions \( G_1, G_2, \ldots, G_\sigma \) operate in parallel and independently of one another, as depicted in Fig. 2.

The security proof is elaborated for a TI of order \( \gamma = 1 \) that operates on \( \sigma_{\text{in}} = \sigma_{\text{out}} = 3 \) shares, but can trivially be generalized to cover other parameter values. Arribas et al. \cite{2} adopt the same view of data-dependent propagation delays as Li et al. \cite{14}, which was illustrated in Eq. (1), and let \( D_{G_i}(X_2, X_3, Y_1) \).

\(^3\) The contents of the FDTC 2018 article \cite{2} also appear in the PhD thesis of Arribas \cite{1}. Although differences are small and solely editorial, this PhD thesis was published after our refutation of the proof was made public. To be precise: an initial draft of our article was forwarded to Arribas, De Cnudde, and Šijačić on April 9, 2020 and sparked a discussion that ended on April 27, 2020, whereas the PhD thesis appeared online in May 2020. The PhD thesis is not further cited to exclude dependencies on our work with 100% certainty, which will be particularly important for Section 3.1.
$D_{G_1}(X_1, X_3, Y_2)$, and $D_{G_3}(X_1, X_2, Y_3)$ 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., $(\chi_1, \chi_2, \chi_3) \triangleq (0, 0, 0)$, and (ii) the output shares $(Y_1, Y_2, Y_3)$ are the result of evaluating this constant, i.e., $Y_1 \triangleq G_1(\chi_2, \chi_3)$, given that the inclusion of $Y_1 \triangleq G_1(X_2, X_3)$ would be redundant, and similarly for $Y_2$ and $Y_3$. Without loss of generality, Arribas et al. [2] assume that $D_{G_1} \geq D_{G_2} \geq D_{G_3}$. In this case, $G_1$ is the first component function to produce an erroneous output share, and $G_2$ and $G_3$ supposedly do not affect the fault sensitivity of the TI as a whole. Stated otherwise, the attacker knows whether or not $G_1$ failed, but it cannot be measured or inferred 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 the unmasked secret $X = X_1 \oplus X_2 \oplus X_3$, which ends the proof.

**Revised Version** In the revised version of the proof, De Cnudde [7] 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. In this scenario, an attacker only learns whether the output is correct or not. Assumption 4 is made to preclude the aforementioned variation of FSA by Mischke et al. [15] where identical inputs in subsequent clock cycles are spotted. 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 argues that the attacker model is ill-defined. Section 3.2 points out that the proof makes abstraction of three physical phenomena that can only justifiably be made abstraction of in the original side-channel-resistance setting. Section 3.3 identifies a fatal mathematical 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. Our main concern is that many variations of FSA exist [12, 16, 20], and it is unclear which variations
are covered by the proof. Arribas et al. [2,7] supplemented Assumption 1 with a summary of the original FSA by Li et al. [14], 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 that it is just an example. First, De Cnudde [7] explicitly acknowledges support for the FSA by Mischke et al. [15], which implies that Assumption 1 must be broad enough to cover this type of attack. Second, Arribas et al. [2,7] motivate their work by describing how Moradi et al. [16] successfully attack non-glitch-resistant masking schemes, and then suggest that TIs provide a solution. Based on this backstory, one would expect the FSA by Moradi et al. [16] to be covered. Conversely, an argument for the sole coverage of the original FSA by Li et al. [14] can be made: Arribas et al. [2] only simulate this type of attack in their experiments. Yet again, no definitive conclusions can be drawn. 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 be impractical to test them all.

An aggravating factor to this confusing situation is that the original FSA by Li et al. [14], which serves as a supplement to Assumption 1, does not even apply to TIs. This type of FSA 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 $M_j$. It would thus be pointless for an attacker to gradually increase the fault intensity until an erroneous output appears. A cryptosystem is, obviously and by default, secure against an inapplicable attack; no security proof is needed to confirm this. Hence, unless Arribas et al. [2] intended to cover several FSA variations, which cannot be confirmed due to the ambiguous wording, their proof lacks existential motivation. To draw an analogy: just like no paper is needed to prove that a cryptosystem with an immutable key is secure against related-key attacks, no paper is needed to prove that TIs are secure against the original FSA by Li et al. [14]. Moreover, non-glitch-resistant masking schemes, which do not meet the incompleteness requirement in Definition 3 also exhibit ever-changing fault sensitivities, thereby rendering the original FSA by Li et al. [14] equally inapplicable. Hence, unless Arribas et al. [2] intended to cover several FSA variations, the suggested notion that TIs are superior to non-glitch-resistant masking schemes is unsupported.

Arribas et al. [2,7] do not acknowledge that the original FSA by Li et al. [14] is inapplicable to TIs and other masking schemes. In fact, the opposite is suggested. In their experiments, Arribas et al. [2] are able to (unsuccessfully) attack TIs using a simulated version of the FSA by Li et al. [14]. This simulated version is mentioned to be unrealistically in favor of the attacker, who receives exact, noiseless values of the maximum propagation delay $D_{(G_1,G_2,G_3)} = \max(D_{G_1}, D_{G_2}, D_{G_3})$ rather than noisy estimates. This is reasonable: the only drawback of noise is that more measurements are needed for an identical attack to succeed. However, arguably the most unrealistic aspect is unacknowledged: in actual, real-world attacks, not even a noisy estimate of $D_{(G_1,G_2,G_3)}$ could be obtained owing to the
ever-changing fault sensitivities. For given input shares \((X_1, X_2, X_3)\), only the binary outcome of a single comparison \(D(G_1, G_2, G_3) \lessgtr D'\), where \(D'\) relates to the fault intensity, is obtainable. Given that Arribas et al. \[2\] drastically alter the physical reality of the FSA by Li et al. \[14\], the uncertainty on which FSA variations are covered by the proof increases. It would be counterintuitive to approve a fully artificial, physics-defying alteration but disapprove less disruptive FSA variations \[12,13,15,16\] that actually abide physical laws.

In our analysis, we try to work around the uncertainties. Initially, in Sections 3.2 to 3.3 the only FSA variation we consider to be covered is the original one by Li et al. \[14\], even though applying this technique to TIs is physically impossible. Upon pointing out a fatal mathematical error in the proof and providing an example of a TI that succumbs to FSA within this physically impossible context, we consider the proof to be officially refuted. Afterwards, in Section 3.4 we contemplate FSA variations of which the application to TIs is physically plausible \[12,15\]. We show that the mathematical error continues to exist in this new context, and provide examples of TIs that succumb to FSA. There is no guarantee that the attacks in Section 3.4 were meant to be covered by Arribas et al. \[2,7\], but we provide them anyway: members of academia and industry are primarily interested in attacks that have implications to the real world. It should be noted that all the aforementioned FSA variations \[12–16\] are approximately equally difficult to perform from a technological perspective, i.e., differences lie in query strategies and the data processing rather than in the cost of the equipment. Therefore, any proposed FSA countermeasure should provide a broad coverage in order to be adequate for industrial adoption. Our analysis shows that TIs are unpromising in this regard.

Lastly, we identify one ambiguity for each assumption made in Section 2.4. In contrast to the previously discussed uncertainty about the coverage and the physical plausibility of several FSA variations, all four ambiguities can easily be mitigated by making a conservative assumption. For Assumption 1 the notion of propagation delays \(D\) being “measured” is open to interpretation. Through fault-injection methods such as heating and under-powering, propagation delays \(D\) are not only measured but also increased in possibly complex, non-linear ways. To be conservative, we further only consider reductions of the clock period \(T_{clk}\), as \(D\) then remains unaltered. For Assumption 2 it is unclear whether the suggested notion of parallelism is in conflict with local fault-injection methods. Schellenberg et al. \[20\] previously performed FSA using a laser, for example. Again, we are conservative by only considering reductions of the clock period \(T_{clk}\), i.e., a global fault-injection method. For Assumption 3 De Cnudde \[7\] does not comment on undetectable faults. Hence, the consequences of, for example, identical faults in a duplicated cipher implementation are unclear. Depending on the chosen comparison method, the same concern might apply if fault injection results in erroneous output shares \((y_1 \oplus e_1, y_2 \oplus e_2, y_3 \oplus e_3)\) such that \(e_1 \oplus e_2 \oplus e_3 = 0\). 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. To be conservative, we consider it a secret.
3.2 Deficiencies of The Physical Model

As mentioned earlier-on, the strength of TIs is that few assumptions about the underlying hardware are made: physical phenomena such as glitches, noise, and data-dependent propagation delays can easily be made abstraction of when proving resistance to side-channel attacks \[4,17\] as claimed in Theorem \[2\] Arribas et al. \[2,7\] continue this tradition by making almost equally strong abstractions in their FSA-resistance proof, but the result is less convincing. For static CMOS logic in particular, we argue that basic physical properties of gate propagation delays cause anomalies in the proof. First of all, we show that the component-wise delay functions \(D_{G_i}\) as defined in Section \[2.4\] should be redefined in order to capture unforeseen data dependencies. Subsequently, we point out that in the presence of glitches and noise, the constraint \(D_{G_1} \geq D_{G_2} \geq D_{G_3}\) does not preclude \(D_{G_2}\) and \(D_{G_3}\) from being measured. Hence, the independence of input share \(X_1\) cannot be enforced. To avoid three-share dependencies, a collaborating attacker is required, which is not usually how such a person or organization behaves in the real world. Although the described anomalies will be overshadowed in Section \[3.3\] by an unrelated, purely mathematical flaw that refutes the proof single-handedly, we still forewarn potential follow-up works that physical abstractions are not without pitfalls.

Data-Dependent Propagation Delays Are Everywhere

The problem of data-dependent propagation delays is more severe than Li et al. \[14\] and Arribas et al. \[2\] assume. In addition to data dependencies caused by an unbalanced network of gates, as discussed in Section \[2.2\] data dependencies also arise within a single, isolated gate, given that gate propagation delays cannot accurately be described by a single parameter \(\delta\). For static CMOS logic, the latter dependencies 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. \[19, Chapter 6\], 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 \(C_{\text{load}}\) 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 \(C_{\text{int}}\) 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 \(D = 0\).

Consequentially, the common distinction between insecure AND/NAND/ OR/NOR gates and secure XOR/XNOR gates \[2,10,13\] 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 $A$. For the FSA-resistance proof of Arribas et al. [2], the following problem emerges: the componentwise delay functions $D_{G_1}(X_2, X_3, Y_1)$, $D_{G_2}(X_1, X_3, Y_2)$, and $D_{G_3}(X_1, X_2, Y_3)$ cannot capture all possible delay dependencies. As component functions $G_i$ are usually non-injective, distinct reset values $(\chi_2, \chi_3)$ can map to the same $Y_1 \triangleq G_1(\chi_2, \chi_3)$, yet result in different charges on the internal circuit nodes and thus different propagation delays $D_{G_1}$, and similarly for $G_2$ and $G_3$. To solve this problem, we redefine the componentwise delay functions in Eq. (1), where superscripts (1) and (2) refer to previously and newly stored values by the input registers respectively. For the revised proof of De Cnudde [7], it suffices to alter Eq. (4) such that the reset value is uniformly distributed rather than constant.

Fig. 3: A two-input NAND gate in static CMOS technology [19, Chapter 6]. (a) The circuit, which consists of two nMOS transistors in series and two pMOS transistors in parallel. (b) An RC model, where the load capacitance $C_{\text{load}}$ comprises an aggregate of all gates driven by the NAND gate.
Table 1: Data-dependent propagation delays of the two-input NAND gate shown in Fig. 3 as simulated by Rabaey et al. [19, 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.

<table>
<thead>
<tr>
<th>Input A</th>
<th>Input B</th>
<th>Output C</th>
<th>Delay D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 → 1</td>
<td>0 → 1</td>
<td>1 → 0</td>
<td>69 ps</td>
</tr>
<tr>
<td>1 → 1</td>
<td>0 → 1</td>
<td>1 → 0</td>
<td>62 ps</td>
</tr>
<tr>
<td>0 → 1</td>
<td>1 → 0</td>
<td>1 → 0</td>
<td>50 ps</td>
</tr>
<tr>
<td>1 → 0</td>
<td>1 → 0</td>
<td>0 → 1</td>
<td>35 ps</td>
</tr>
<tr>
<td>1 → 0</td>
<td>0 → 1</td>
<td>0 → 1</td>
<td>76 ps</td>
</tr>
<tr>
<td>1 → 0</td>
<td>1 → 0</td>
<td>0 → 1</td>
<td>57 ps</td>
</tr>
</tbody>
</table>

\[ D_{G_1}(X_2^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}), \]
\[ D_{G_2}(X_1^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}), \]
\[ D_{G_3}(X_1^{(1)}, X_2^{(1)}, X_2^{(2)}), \]

where \((X_1^{(1)}, X_2^{(1)}, X_3^{(1)}) \neq (X_1, X_2, X_3). \quad (4)\]

The Danger of Glitches Consider a three-share TI with single-bit outputs \(Y_i\) that respond to a given transition of the input shares \(X_i\) as shown in Fig. 4a. As assumed in the FSA-resistance proof, \(D_{G_1} \geq D_{G_2} \geq D_{G_3}\). For simplicity, we chose \(D_{G_3} = 0\) and do not further consider the corresponding output node. More importantly, the output node of component function \(G_1\) exhibits a glitch consisting of two transitions such that the output \(Y_1\) is correct in an interval around \(D_{G_2}\). Hence, the attacker can measure not only \(D_{G_1}\) but also \(D_{G_2}\), thereby utilizing a three-share dependency that potentially reveals information on the unshared secret \(X \triangleq X_1 \oplus X_2 \oplus X_3\). This is under the implicit assumption of Arribas et al. [2,7] that an attacker is not bothered by the ever-changing masks \(M_i\), which was argued to be physically impossible in Section 3.1.

An attacker who is kind enough to execute the original FSA by Li et al. [14] such that the clock period \(T'_{clk}\) is decreased in (infinitely) small steps starting from the nominal value \(T_{clk}\), evidently, only obtains \(D_{G_1}\). Note that in a more efficient binary search for \(D_{G_1}\), an attacker might accidentally overshoot the glitch of \(Y_1\) and end up with a measurement of \(D_{G_2}\). Even more problematic, a real-world attacker does not make gestures of goodwill and would measure both \(D_{G_1}\) and \(D_{G_2}\) on purpose. Forsaking support for glitches is not a satisfactory solution to this problem: TIs are specifically advertised as a glitch-resistant masking scheme [4,17] and thus lose their competitive edge over other masking schemes for logic that is devoid of glitches. Also remark that for FSA variations in which the fault intensity is not gradually increased, in case these are covered as previously discussed in Section 3.1, propagation delays cannot unambiguously be defined using a single variable \(D\). For an output node that exhibits a glitch, not only the last transition but also one or more earlier transitions then impact the
security, and the meaning of the proof becomes unclear. This ambiguity becomes increasingly unsustainable with an increasing number of edges that constitute the glitch.

Noise and Time-Variant Fault Sensitivity

Unfortunately, electronic circuits are subject to noise \[19\], which we define as irreproducible deviations from the nominal behavior caused by randomly moving particles. For example, the resistive 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 \(X(1) \rightarrow X(2)\), the propagation delay \(D\) of an output node is more accurately described by a Gaussian-like distribution than by constant. Second, if the setup and/or hold time of a flip-flop that samples an output bit \(Y\) is violated, it enters a metastable state that eventually resolves to either 0 or 1 depending on noise sources within this flip-flop.

For the FSA-resistance proof, the lack of a clear-cut threshold \(D_G\) separating correct and faulty outputs results in the following anomaly. Consider a three-share TI with single-bit outputs \(Y_i\) that respond to a given transition of the input shares \(X_i\) as shown in Fig. 4b. Again, \(D_{G_1} \geq D_{G_2} \geq D_{G_3}\) and \(D_{G_3} = 0\). For the first two output nodes, the probability of registering a 1 has sloped edges due to noise. For simplicity, these slopes are drawn as straight lines; more accurate curves \[19\] do not change our reasoning. An attacker is not precluded from measuring the probability of sampling a correct output \((Y_1, Y_2, Y_3)\), i.e., \(P_{\text{correct}, G_1, G_2, G_3} \triangleq \prod_{i=1}^{3} P_{\text{correct}, G_i}\), which results in a three-share dependency for some intervals of the reduced clock period \(T'_{\text{clk}}\). Again, cooperation from the attacker is required to resist the original FSA by Li et al. \[14\], under the implicit assumption of Arribas et al. \[2, 7\] that the ever-changing masks \(M_i\) are not a
problem. Furthermore, due to the slopes, propagation delays $D$ are hard to define using a single variable, so the meaning of the proof becomes unclear.

### 3.3 The Incompleteness Fallacy

We now make abstraction of the physical deficiencies in Section 3.2 and assess the FSA-resistance proof from a purely mathematical perspective. A first peculiarity of the proof is that the made assumptions are not explicitly incorporated. For example, 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 such formalizations are missing as a stepping stone, a fatal flaw arises, as discussed next.

For now, we still assume that an attacker can obtain the exact value of a TI’s propagation delay $D_{(G_1,G_2,G_3)}$ for any given transition of the input shares ($X_1, X_2, X_3$). As argued in Section 3.1, this is physically impossible, but it complies with the experiments of Arribas et al. 2 and the reasoning behind their proof. 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 $D_{(G_1,G_2,G_3)} = \max(D_{G_1}, D_{G_2}, D_{G_3})$ depends on all three input shares and thus also reveals information on all three input shares. Even though the value of $D_{(G_1,G_2,G_3)}$ is taken from a single component function $G_i$, e.g., $D_{(G_1,G_2,G_3)} = D_{G_i}$ if $D_{G_1} \geq D_{G_2} \geq D_{G_3}$, the three-share dependency remains present. For componentwise delays as defined in Eq. \ref{eq:5}, it can be seen in Eq. \ref{eq:6} that constraints involving all three input shares $X_i$ are implied, thereby possibly revealing information on the unshared secret $X \equiv X_1 \oplus X_2 \oplus X_3$.

\[
D_{(G_1,G_2,G_3)} = \alpha \quad \implies \quad D_{G_1}(X_2^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}) \in [0, \alpha], \quad \text{Eq. (5)}
\]

Consider, for example, a Hamming weight (HW) model of the componentwise delays $D_{G_i}$ as specified in Eq. \ref{eq:5}. We let the input shares $X_i$ initially be zero, which is a typical reset value for registers. The unit of measurement of the $D_{G_i}$’s is arbitrary and is, therefore, omitted. Also, the function $G(X)$ and its associated component functions $G_1(X_2, X_3)$, $G_2(X_1, X_3)$, and $G_3(X_1, X_2)$ are arbitrary and thus unspecified. As it turns out, the expected value $E[D_{(G_1,G_2,G_3)}]$ depends on the unmasked characteristic $HW(X^{(2)})$. For a nibble size $\lambda = 2$, it holds that $E[D_{(G_1,G_2,G_3)}]$ equals 2.625, 2.5625, and 2.5 if $HW(X^{(2)})$ equals 0, 1, and 2 respectively. These numbers can be derived by exhaustively evaluating all $2^{6 \lambda} = 64$ possible input transitions, as we did in MATLAB. Note that in a proof-of-concept of the original FSA by Li et al. \cite{14}, a similar HW exploit is used. Also remark that this refutation can be made applicable to the revised proof of De Cnudde \cite{7} by removing the constraint on the reset value in Eq. \ref{eq:5}. 


For Arribas et al. remains present. Consider the set \( X \) less informative than the complete value relates to the fault intensity. Although the single-bit result of this comparison is the case in practice, the TI is vulnerable to FSA. Note that if Eq. (8) is true, a resides. Hence, unless the condition in Eq. (8) is true, which is unlikely to be each evaluation, the attacker knows in which of the two sets the actual transition of the input shares (\( \sigma \lambda \)) of the input shares (\( \sigma \lambda \)). As illustrated in Fig. 5, for any given fault sensitivity threshold \( D' \), the set \( X_{\text{trans}} \) can be partitioned into the two subsets that are defined in Eq. (7). 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. (8) is true, a similar condition for \( X_{\text{trans, faulty}}(D') \) is also true.

\[
D_{G_1}(X_2^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}) \triangleq \text{HW}(X_2^{(2)}) + \text{HW}(X_3^{(2)}), \\
D_{G_2}(X_1^{(1)}, X_1^{(2)}, X_2^{(1)}, X_3^{(2)}) \triangleq \text{HW}(X_1^{(2)}) + \text{HW}(X_3^{(2)}), \\
D_{G_3}(X_1^{(1)}, X_2^{(1)}, X_1^{(2)}, X_2^{(2)}) \triangleq \text{HW}(X_1^{(2)}) + \text{HW}(X_2^{(2)}),
\]

where

\[
X_1^{(1)} \triangleq X_2^{(1)} \triangleq X_3^{(1)} \triangleq 0.
\]

3.4 Physically Plausible Attacks

We now consider the FSA-resistance proof [2,7] in the context of physically plausible attacks [12,15]. Although such attacks were not necessarily intended to be covered by the proof, we solidify that the proof provides no security guarantees even for a weaker attacker who is bothered by the ever-changing masks \( M_i \). First, we show that the incompleteness fallacy from Section 3.3 still applies to this weaker attacker model. Subsequently, we specify an instance of the componentwise delays \( D_{G_3} \) in Eq. (4) that succumbs to FSA, both for the original proof by Arribas et al. [2] and for the hardened proof by De Cnudde [7], in which Assumption 4 respectively does not and does exist. As noise and glitches cannot adequately be captured by Eq. (4), as argued in Section 3.2, we still make abstraction of these phenomena just like Arribas et al. [2,7].

The Incompleteness Flaw Revisited Recall that for any given transition of the input shares \( (X_1, X_2, X_3) \) in a physically plausible FSA, only the binary outcome of a single comparison \( D_{(G_1, G_2, G_3)} \) \( \leq D' \) can be observed, where \( D' \) relates to the fault intensity. Although the single-bit result of this comparison is less informative than the complete value \( D_{(G_1, G_2, G_3)} \), the three-share dependency remains present. Consider the set \( X_{\text{trans}} \) of all possible shared input transitions. For Arribas et al. [2], the cardinality \( |X_{\text{trans}}| = 2^{6 \lambda} \), for De Cnudde [7], \( |X_{\text{trans}}| = 4^{\sigma \lambda} \). As illustrated in Fig. 5, for any given fault sensitivity threshold \( D' \), the set \( X_{\text{trans}} \) can be partitioned into the two subsets that are defined in Eq. (7). For each evaluation, the attacker knows in which of the two sets the actual transition resides. Hence, unless the condition in Eq. (8) is true, which is unlikely to be the case in practice, the TI is vulnerable to FSA. Note that if Eq. (8) is true, a similar condition for \( X_{\text{trans, faulty}}(D') \) is also true.

\[
X_{\text{trans, correct}}(D') \triangleq \{ (X_1^{(1)}, X_2^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}) \in X_{\text{trans}} | \\
(D_{G_1}(X_2^{(1)}, X_3^{(1)}, X_2^{(2)}, X_3^{(2)}) < D') \land (D_{G_2}(X_1^{(1)}, X_3^{(1)}, X_1^{(2)}, X_3^{(2)}) < D') \land \\
(D_{G_3}(X_1^{(1)}, X_2^{(1)}, X_1^{(2)}, X_2^{(2)}) < D') \},
\]

\[
X_{\text{trans, faulty}}(D') \triangleq X_{\text{trans}} \setminus X_{\text{trans, correct}}(D').
\]

\[
\forall D' \in [0, T_{\text{clk}}], \forall X \in \{0, 1\}^{6 \lambda}, \\
|\{ (X_1^{(1)}, X_2^{(1)}, X_3^{(1)}, X_1^{(2)}, X_2^{(2)}, X_3^{(2)}) \in X_{\text{trans, correct}}(D') | \\
X_1^{(2)} \oplus X_2^{(2)} \oplus X_3^{(2)} = X \}| = |X_{\text{trans, correct}}(D')|/2^{6 \lambda}.
\]
Excluding Assumption 4 Consider an arbitrary invertible, quadratic function $G : \{0, 1\}^\lambda \rightarrow \{0, 1\}^\lambda$, which can be thought of as an S-box. For its arbitrary TI, let the componentwise delays $D_{G_1}$, $D_{G_2}$, and $D_{G_3}$ be zero if the respective function outputs remain unchanged, as specified in Eq. (9). This behavior is consistent with static CMOS logic, as previously discussed for the NAND gate in Fig. 3. If the output of a $G_i$ changes, let the delay be equal to an arbitrary strictly positive constant. 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 $3 \cdot 4^\lambda$ hardwired bits and are thus area-inefficient for typical values of $\lambda$, e.g., $\lambda \in \{4, 8\}$, but we are free to adopt any piece of hardware that complies with the terms of the proof.

\[
D_{G_1} \triangleq \begin{cases} 
0 & \text{if } G_1(X_2^{(2)}, X_3^{(2)}) = G_1(0, 0) \\
3 & \text{otherwise},
\end{cases}
\]

\[
D_{G_2} \triangleq \begin{cases} 
0 & \text{if } G_2(X_1^{(2)}, X_3^{(2)}) = G_2(0, 0) \\
3 & \text{otherwise},
\end{cases}
\]

\[
D_{G_3} \triangleq \begin{cases} 
0 & \text{if } G_3(X_1^{(2)}, X_2^{(2)}) = G_3(0, 0) \\
3 & \text{otherwise},
\end{cases}
\]

Due to the independency of component functions, $D_{(G_1, G_2, G_3)} \triangleq \max(D_{G_1}, D_{G_2}, D_{G_3})$. Hence, we obtain Eq. (10) from Eq. (9).
given that the TI then realizes a permutation of the set
\{\mathbf{Y}_{\text{output shares}}\} such that the threshold between a correct and a faulty computation, the data-dependent statistic in Eq. (12) arises. A correct output implies that the threshold between a correct and a faulty computation, the data-dependent statistic in Eq. (12) arises. A correct output implies that the threshold between a correct and a faulty computation, the data-dependent statistic in Eq. (12) arises. A correct output implies that the threshold between a correct and a faulty computation, the data-dependent statistic in Eq. (12) arises.

\[ D_{(G_1, G_2, G_3)} = \begin{cases} 0 & \text{if } (X_1^{(2)}, X_2^{(2)}, X_3^{(2)}) \in D_0 \\ 3 & \text{otherwise,} \end{cases} \]

where \( D_0 = \{(X_1, X_2, X_3) \mid (G_1(X_2, X_3) = G_1(0, 0)) \land (G_2(X_1, X_3) = G_2(0, 0)) \land (G_3(X_1, X_2) = G_3(0, 0))\}. \]

A necessary but insufficient condition for \( D_{(G_1, G_2, G_3)} = 0 \) is given in Eq. (11). It follows that \( |D_0| \leq 2^{\lambda (\sigma - 1)} \). As \((0, 0, 0) \in D_0\), it also holds that \(|D_0| \geq 1\). The exact value of \( |D_0| \in [1, 2^{\lambda (\sigma - 1)}] \) can easily be computed for a given TI. If the output shares \( \mathbf{Y}_i \) are uniform according to Definition 1, it holds that \(|D_0| = 1\), given that the TI then realizes a permutation of the set \{0, 1\}^\lambda.

\[ (X_1^{(2)}, X_2^{(2)}, X_3^{(2)}) \in D_0 \] \[ \implies G_1(X_2^{(2)}, X_3^{(2)}) \oplus G_2(X_1^{(2)}, X_3^{(2)}) \oplus G_3(X_1^{(2)}, X_2^{(2)}) = G_1(0, 0) \oplus G_2(0, 0) \oplus G_3(0, 0) \] \[ \implies G(X^{(2)}) = G(0) \implies X^{(2)} = 0. \]

If the attacker reduces the clock period \( T_{\text{clk}} \) such that \( D_{(G_1, G_2, G_3)} = 2 \) is the threshold between a correct and a faulty computation, the data-dependent statistic in Eq. (12) arises. A correct output implies that \( X^{(2)} = 0 \), i.e., a secret value is revealed in its entirety. Observe that the presented attack is similar to the aforementioned FSA variation where identical inputs in subsequent clock cycles are spotted \cite{12, 15}; the difference lies in the additional complexity of component functions \( G_i \) being non-injective. Note also that the specification of the componentwise delays in Eq. (9) can be relaxed to accommodate a more realistic attack. For the ‘otherwise’ cases in Eq. (9), \( D_{G_i} \) 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.

\[ P(D_{(G_1, G_2, G_3)} < 2) = \begin{cases} |D_0|/2^{\lambda (\sigma - 1)} & \text{if } X^{(2)} = 0 \\ 0 & \text{otherwise.} \end{cases} \]

To further quantify the practicality of the above attack, we simulate the timing behavior of a TI of a quadratic, invertible, 4 \( \times \) 4 S-box that is mapped to a Xilinx Spartan-6 field-programmable gate array (FPGA) with a 1 ps resolution. Specifications of the original function \( G(X) \) and the first component function of its TI, \( G_1(X_2, X_3) \), are taken from Poschmann et al. \cite{18} and are repeated here in Eq. (13) and Eq. (14) respectively. Similarly, \( G_2(X_1, X_3) \triangleq G_1(X_3, X_1) \) and \( G_3(X_1, X_2) \triangleq G_1(X_1, X_2) \). In Xilinx Integrated Synthesis Environment (ISE), we created a test bench that applies all \( 2^4 \) and \( 2^{3\lambda} \) possible input transitions to \( G \) and its TI respectively, where \( 0 \) is the reset value. By feeding a post-place \& route model into ISE Simulator (ISim), all respective propagation delays \( D_G \) and \( D_{(G_1, G_2, G_3)} \) are obtained. Except for the \( 0 \rightarrow 0 \) transition, all delays are in the
6 ns to 8 ns interval; no glitches are observed. In Fig. 6a we plot the difference of probabilities $P(D_G \leq D | X) - P(D_G \leq D)$ as a function of $D$ for all $X \in \{0, 1\}^\lambda$, and similarly for $D(G_1, G_2, G_3)$ in Fig. 6b. Instead of $2^\lambda = 16$ coinciding curves, as in the FSA-secure case, substantially differing curves appear in each plot. The $0 \rightarrow 0$ transition in the 0 ns to 6 ns interval is the most easily exploitable as neither a precise delay model nor precise control over the reduced clock period $T'_{\text{clk}}$ nor a noiseless environment is required to succeed, yet the whole 6 ns to 8 ns interval is worrisome as well.

$$y_3 \triangleq x_2 \oplus x_1 \oplus x_0 \oplus x_3 x_0,$$
$$y_2 \triangleq x_3 \oplus x_1 x_0,$$
$$y_1 \triangleq x_2 \oplus x_1 \oplus x_3 x_0,$$
$$y_0 \triangleq x_1 \oplus x_2 x_0.$$  \hspace{1cm} (13)

$$y_{3,1} \triangleq x_{2,2} \oplus x_{1,2} \oplus x_{0,2} \oplus x_{3,2} x_{0,2} \oplus x_{3,2} x_{0,3} \oplus x_{3,3} x_{0,2},$$
$$y_{2,1} \triangleq x_{3,2} \oplus x_{1,2} x_{0,2} \oplus x_{1,2} x_{0,3} \oplus x_{1,3} x_{0,2},$$
$$y_{1,1} \triangleq x_{2,2} \oplus x_{1,2} \oplus x_{3,2} x_{0,2} \oplus x_{3,2} x_{0,3} \oplus x_{3,3} x_{0,2},$$
$$y_{0,1} \triangleq x_{1,2} \oplus x_{2,2} x_{0,2} \oplus x_{2,2} x_{0,3} \oplus x_{2,3} x_{0,2}. \hspace{1cm} (14)$$

![Fig. 6: Simulated delay characteristics of the 4 × 4 S-box in Eq. (13) mapped to a Xilinx Spartan-6 FPGA. To be FSA-secure, all $2^4 = 16$ curves should coincide with the horizontal axis, i.e., a constant zero.](image)

Including Assumption Consider an arbitrary invertible, affine function $G : \{0, 1\}^\lambda \rightarrow \{0, 1\}^\lambda$. Its TI operates on $\sigma_{\text{in}} = \sigma_{\text{out}} = 2$ shares and is constructed as follows: $G_1(x_1) \triangleq G(x_1)$ and $G_2(x_2) \triangleq G(x_2)$. The componentwise delays $D_{G_i}$ are given in Eq. (15), 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 \( G = G_1 = G_2 \) is injective, it is reasonable to assume that \( D_G = 0 \) if and only if input share \( X_i \) remains unchanged. Furthermore, for one particular nonvoid transition, the strictly positive delay is particularly small.

Due to the independency of component functions, \( D_{(G_1, G_2)} \triangleq \max(D_{G_1}, D_{G_2}) \).

Again, the attacker reduces the clock period \( T_{\text{clk}} \) such that \( D_{(G_1, G_2)} = 2 \) is the threshold between a correct and a faulty computation. Through the tree diagram in Fig. 7, we obtain the compromising statistic in Eq. (16), i.e., the probability that the output is correct is slightly higher if \( X^{(2)} = 0 \). To independently verify the correctness of Eq. (16), for all \( \lambda \in [1, 6] \), we let a MATLAB script evaluate Eq. (15) and \( D_{(G_1, G_2)} \) for all \( 4^\sigma \lambda \) possible input transitions.

\[
P\left(D_{(G_1, G_2)} < 2\right) = \begin{cases} 
2^{-\lambda} \sigma \left(1 + 3 \cdot 2^{-\lambda (\sigma - 1)}\right), & \text{if } X^{(2)} = 0 \\
2^{-\lambda} \sigma \left(1 + 2 \cdot 2^{-\lambda (\sigma - 1)}\right), & \text{otherwise.} 
\end{cases} \tag{16}
\]

Admittedly, the theoretical exploit in Eq. (16) is not as practically viable as for the original proof in Eq. (12). Nevertheless, it is important to acknowledge that increasing a TI’s intake of random bits is not necessarily worth its associated cost. Such resources can also be spend on countermeasures against physical attacks that are supported by a correct security proof.

4 Concluding Remarks

To conclude this article, we provide two recommendations for future work. A first, general recommendation is that mathematical proofs should be formal in order to protect oneself from wishful thinking. If Arribas et al. [2,7] would have complemented the informal proof in Section 2.4 with equations, they likely would have discovered the fatal three-share dependency in Eq. (5) themselves. A second, more specific recommendation is to further investigate the implications of the data-dependent gate propagation delays in Section 3.2. Several FSA countermeasures that rely on the balancing of gate depth levels [9,10,13] are seemingly insecure in this light; the case \( D = 0 \) is particularly worrisome.
Fig. 7: Tree diagram representing the distribution of $D_{G_1, G_2} \triangleq \max(D_{G_1}, D_{G_2})$, where $D_{G_1}$ and $D_{G_2}$ are defined in Eq. (15). Note that $\sigma = 2$. 

<table>
<thead>
<tr>
<th>$X^{(2)}$</th>
<th>$(X^{(2)}_1, X^{(2)}_2)$</th>
<th>$(X^{(1)}_1, X^{(1)}_2)$</th>
<th>$D_{(G_1, G_2)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X^{(1)}$</td>
<td>$(0, 0)$</td>
<td>$(0, 0)$</td>
<td>$2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td></td>
<td>$(0, 1)$</td>
<td>$(0, 1)$</td>
<td>$2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td></td>
<td>$(1, 0)$</td>
<td>$(1, 0)$</td>
<td>$2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td></td>
<td>$(1, 1)$</td>
<td>$(1, 1)$</td>
<td>$2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1 - 4 \cdot 2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td>$(x_1, x_2)$, where $x \neq 0$</td>
<td>$(x_1, x_2)$, where $x \neq 0$</td>
<td>$(x_1, x_2)$, where $x \neq 0$</td>
<td>$2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1 - 2 \cdot 2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td>$(x_1, x_2)$, where $x_2 \notin {0, x_1}$ and $x_2 \oplus x_3 = x_1$</td>
<td>$(x_2, x_3)$, where $x_2 \notin {0, x_1}$ and $x_2 \oplus x_3 = x_1$</td>
<td>$(x_2, x_3)$, where $x_2 \notin {0, x_1}$ and $x_2 \oplus x_3 = x_1$</td>
<td>$2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1 - 2 \cdot 2^{-\lambda \sigma}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1 - 2 \cdot 2^{-\lambda \sigma}$</td>
</tr>
</tbody>
</table>
References


