SoK: Parameterization of Fault Adversary Models Connecting Theory and Practice

Dilara Toprakhisar*[1], Svetla Nikova[1], and Ventzislav Nikov[2]

[1] COSIC, KU Leuven, Leuven, Belgium,
firstname.lastname@esat.kuleuven.be
[2] NXP Semiconductors, Leuven, Belgium
venci.nikov@gmail.com

Abstract. Since the first fault attack by Boneh *et al.* in 1997, various physical fault injection mechanisms have been explored to induce errors in electronic systems. Subsequent fault analysis methods of these errors have been studied, and successfully used to attack many cryptographic implementations. This poses a significant challenge to the secure implementation of cryptographic algorithms. To address this, numerous countermeasures have been proposed. Nevertheless, these countermeasures are primarily designed to protect against the particular assumptions made by the fault analysis methods. These assumptions, however, encompass only a limited range of the capabilities inherent to physical fault injection mechanisms.

In this paper, we narrow our focus to fault attacks and countermeasures specific to ASICs, and introduce a novel parameterized fault adversary model capturing an adversary's control over an ASIC. We systematically map (a) the physical fault injection mechanisms, (b) adversary models assumed in fault analysis, and (c) adversary models used to design countermeasures into our introduced model. This model forms the basis for our comprehensive exploration that covers a broad spectrum of fault attacks and countermeasures within symmetric key cryptography as a comprehensive survey. Furthermore, our investigation highlights a notable misalignment among the adversary models assumed in countermeasures, fault attacks, and the intrinsic capabilities of the physical fault injection mechanisms. Through this study, we emphasize the need to reevaluate existing fault adversary models, and advocate for the development of a unified model.

Keywords: Adversarial Models · Fault Attacks · Fault Countermeasures.

1 Introduction

The first fault attack by Boneh *et al.* [10] initiated a new research area focused on the malicious injection of faults and their mathematical analyses to attack

* Corresponding author.

cryptographic implementations. This seminal milestone also instigated the development of countermeasures to mitigate these attacks. Instead of targeting the cryptanalytic properties of the algorithms, these attacks exploit implementation vulnerabilities caused by errors. Unlike passive implementation attacks that solely observe the target device's behavior, fault attacks actively disturb computations through physical means, such as clock/voltage glitches [2,4], electromagnetic waves [39], and laser injection [54]. The attacker then observes the device's reaction to the injected faults. Along with the discovered physical fault injection mechanisms, several fault analysis methods analyzing the injected faults have been proposed, including Differential Fault Analysis (DFA) [8], Statistical Ineffective Fault Attacks (SIFA) [21,19], and others. The combination of injecting faults through physical fault injection mechanisms and the subsequent fault analyses has proven successful in real-world scenarios. In parallel to these attacks, numerous countermeasures have been proposed to protect against them. These countermeasures often employ some kind of redundancy (*i.e.,* time, area, or information) to achieve error detection or correction. Besides fault attacks, the emergence of combined attacks that exploit both side-channel and fault vulnerabilities simultaneously necessitates more sophisticated countermeasures capable of mitigating these attacks. In the context of fault attacks, the term *adversarial model* pertains to defining an adversary performing fault injection through physical fault injection mechanisms. Fault analysis methods, as the second step in a fault attack, rely on certain assumptions regarding the injected fault(s). These assumptions formulate an adversary who carries out the fault injection step, ensuring that the faults align with the assumptions. These assumptions encompass factors such as the fault location on the target device and how they alter the target variables. Similarly, countermeasures rely on analogous assumptions to describe the adversary they aim to protect against.

Physical fault injection mechanisms can execute various fault injection scenarios with varying fault locations, number of faults, and so on, which are then exploited by different fault analysis methods. However, as we will show, the proposed fault analysis methods leverage only a fraction of the capabilities offered by these fault injection mechanisms, with each method exploiting specific properties of the errors resulting from the fault injections. Consequently, the divergence among fault analysis methods, each based on different adversarial models and objectives, complicates the comprehensive assessment of the security of cryptographic implementations. Countermeasures proposed in response to this variety of fault analysis methods are, however, tailored to address specific fault analysis methods and adversarial models (*e.g.,* DFA and/or SIFA). Unfortunately, they often fall short of harnessing the capabilities of physical fault injection mechanisms. Recognizing the diverse capabilities of the physical mechanisms, it is crucial to establish more realistic assumptions for countermeasures. This is essential, as fault adversaries possess the potential to exploit a broader spectrum of fault scenarios than previously assumed within the context of fault attacks. Illustratively, Bartkewitz *et al.* [6] demonstrate that an adversary model, typically thought to be challenging to achieve in real-world scenarios, is, in fact, more feasible than previously believed. This finding raises questions about the effectiveness of certain countermeasures.

Inherently protecting against a larger spectrum of adversary models necessitates the development of a consolidated parameterized adversary model encompassing various fault adversary models prevalent in practical contexts. Such a comprehensive model should be capable of accommodating the diverse fault adversary models reflecting the capabilities of physical injection mechanisms. Moreover, it will facilitate a systematic exploration of fault attacks and countermeasures. Such a unified model will enable the designing of countermeasures based on adversaries having a broader and more realistic spectrum of capabilities. Moreover, this approach can contribute to reducing the complexity and the cost of the designed countermeasures by providing a comprehensive and systematic framework to address different physical fault injection mechanisms and adversary models. The literature contains several studies such as [31], [5] and [45] that analyze the theoretical exploitation of injected faults, or formulate a fault adversary model using a range of parameters. However, they often neglect some aspects of a physical adversary, thus failing to provide a comprehensive understanding of how theoretical assumptions (*i.e.,* fault analysis methods and countermeasures) align with practical scenarios. In this work, we establish a novel parameterized fault adversary model comprehensively capturing an adversary's control span on an ASIC, with the goal of assessing the alignment of theoretical assumptions with practical realities. To achieve this objective, we introduce some notions to differentiate between the assumptions inherent in fault analysis methods, countermeasures, and the actual capabilities of a physical adversary. Specifically, we employ the term *physical adversary model* to characterize an adversary physically injecting faults; *analytical adversary model* to characterize an adversary assumed in fault analysis methods; and *mitigative adversary model* to characterize an adversary assumed in countermeasures.

Contributions. In this paper, we investigate the assumptions inherent to fault analysis methods and countermeasures, and discuss their alignment with real-world scenarios. To facilitate this investigation, we first propose a novel parameterized fault adversary model, providing a comprehensive characterization of different factors that a physical fault adversary can control specifically on ASICs. Our model accommodates various adversary capabilities through its comprehensive set of parameters. Then, we employ the introduced parameterized adversary model to describe the impacts of physical fault injection mechanisms on ASICs. We conduct a comparative analysis, presenting both similarities and differences in their respective capabilities, thus offering a comprehensive perspective on real-world feasibility. After describing the capabilities a physical fault adversary can possess in practice, we first present a survey of several fault and combined analysis methods on the *ASIC implementations of symmetric ciphers*, and the countermeasures proposed to mitigate them. We map the analytical adversary models of the presented attacks, and the mitigative adversary models of the countermeasures into the parameterized adversary model. These mappings reveal a discrepancy between the analytical adversary models, mitigative adversary models, and the physical adversary models accommodating the physical fault injection mechanisms. Through this analysis, we reveal certain limitations and challenges of the existing countermeasures against physical fault injection mechanisms. Building upon the mismatch of the different adversarial models and reality, we discuss the shortcomings of the existing analytical adversary models which highlights the need for a unified fault adversary model. In essence, we stress the need to reassess the assumptions underlying the mitigative adversary model, accounting for the broad range of capabilities of the physical fault injection mechanisms. Then, we pose an open question to define a unified adversary model that can be used as a more accurate representation of the fault adversaries and enable researchers to develop more effective countermeasures against fault attacks. We provide suggestions on what such a unified model should contain.

Outline. In Section 2, we discuss the widely used physical fault injection mechanisms and their impacts on ASICs. Then, in Section 3, we introduce a novel parameterized fault adversary model, and in Section 4, we describe the physical fault adversaries using the introduced model. Then, we present a survey of existing fault and combined analysis methods in Section 5, and countermeasures in Section 6 together with the mappings of the respective analytical and mitigative fault models into the parameterized model. Finally, in Section 7, we discuss our findings.

2 Preliminaries

In this section, we describe the physical fault injection mechanisms that are used to inject faults to an ASIC as the first step to attack the implementations of symmetric key algorithms. Additionally, we present the notations that serve as the foundation for our parameterized fault adversary model.

2.1 The Attack Surface: Circuit Model

The attack surface is assumed to be a digital circuit that is formed of gates and wires, where the gates are composed of combinational Boolean logic gates and memory gates. A combinational gate computes its output as a Boolean function of the present inputs. Unlike combinational gates, a memory gate is a clock-synchronized gate where the output depends on the previous input in addition to the present input. In other words, memory gates (*i.e.,* registers), store Boolean variables being dependent on the clock. The digital circuit takes an input, has an internal state, and produces an output where the state corresponds to the secret data stored in the registers. Note that we focus on ASICs and deliberately exclude FPGAs and CPUs since they would necessitate considering also other types of memories such as RAM, ROM, Non-Volatile, etc.

2.2 Physical Fault Injection Mechanisms

In this section, we introduce the most common physical fault injection mechanisms altering the execution of an ASIC: clock glitches [2,36], underpowering and voltage glitches [4,58], EM-fault injections [39,37,38,22], and laser fault injections [54,16,50]. While not delving into the technical details, our focus is solely on elucidating the physical effects of the fault injection mechanisms on ASICs. The efficacy of the mechanisms described in this section has been validated through successful fault attacks against the ASIC implementations of symmetric key algorithms. Among them, non-invasive clock/voltage glitches stand as cost-effective yet powerful methods to inject faults on a global scale on the whole IC. On the other hand, laser fault injection has the highest locality which in turn provides greater precision. Between these two extremes, EM-fault injection impacts the circuits in a particular area chosen by the adversary.

Clock glitches. In synchronous ICs the data is processed by combinational logic blocks separated by memory gates (*i.e.,* D flip-flop registers) sharing the same clock as illustrated in Figure 1. The raising clock edges trigger the registers to latch the data, and in between, the intermediate combinational logic block operates on the data. Once the rising edge of the clock arrives, the signal traveling through the combinational logic block achieves stability. The time taken for the signal to travel through the combinational logic block is called the propagation delay. The set-up time of the register (*i.e.,* the minimum time period the data should remain present at the register before being latched ($t_{\text{set-up}}$)), the maximal propagation delay (*i.e.,* critical path (t_{critical})), the clock skew (δ), and the register delay ($t_{\text{reg'}}$) define the (maximum) clock period T_{clk} of the circuit:

$$T_{\text{clk}} + \delta = t_{\text{critical}} + t_{\text{reg'}} + t_{\text{set-up}}.$$

The propagation delay of computations is susceptible to variations in temperature and power supply voltage. These fluctuations can potentially interrupt the normal functioning of the circuit. Therefore, in order to ensure a reliable circuit operation, the clock period is taken to be greater than T_{clk}.

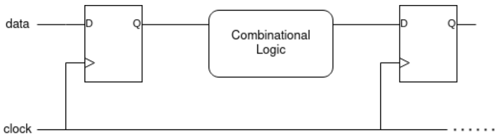

Fig. 1. A synchronously operating IC.

An attacker can alter the external reference clock from which the internal clock of an IC is derived. Such alterations to the external reference clock can allow an attacker to decrease the clock period ($T_{\text{clk'}}$). As $T_{\text{clk'}}$ approaches to t_{critical}, one starts to observe faulty results as the altered clock period prevents the completion of the combinational logic and therefore, the arrival of the correct data at the register on time. As a result, faulty input gets latched in the register. Naturally, decreasing the clock period potentially affects the logical paths that have a propagation delay greater than ($T_{\text{clk'}} - t_{\text{reg'}} - t_{\text{set-up}} + \delta$). Therefore, an attacker lacks direct control over the specific location of the injected fault. In fact, clock glitching can potentially affect the undesired components, leading to undesired faulty outputs. Nevertheless, Ning *et al.* [36] states that the first faulty bit is theoretically located on the critical path characterized by the longest delay. As the fault intensity increases, more bit failures are likely to happen.

Underpowering and voltage glitches. ICs are designed to operate properly within a specified voltage supply range, and any deviation from this specified range may produce faulty outputs. In essence, underpowering and voltage glitches affect the ICs in a similar way to the clock glitches. However, rather than changing the clock period, the decreased supply voltage leads to an increase in the critical path. This is due to the fact the variation in supply voltage amplifies the propagation delay of the gates. Consequently, similar to clock glitches, correct data might not arrive at the register on time. Likewise, undesired components may be affected which will then potentially produce undesired faulty outputs. In addition to manipulating the power supply voltage, an attacker can also alter the critical path through a ground input to the IC.

EM-fault injection. An EM-fault injection directly affects the input and control signals of D flip-flops. In fact, if the fault injection is performed just before the arrival of the rising edge of the clock, then a faulty sampling occurs at the D flip-flop as noted by previous studies [37,38]. As stated by Dumont *et al.* [22], EM-fault injection does not disrupt the interconnect wires.

This technique induces a voltage swing in the IC between the power and ground grid. The falling edge of the swing causes the potential of the clock and input signals to go down. Consequently, the rising edge of the swing triggers the circuit to recover its original state, *i.e.,* all signals start to recover the correct state. However, this causes a race between the clock and input signals [22]. If the clock signal recovers its correct state first, then the register stores a value dictated by the fault injection. The stored faulty value is related to the polarity created by the EM-fault injection; while a positive swing is more likely to cause the register to store 1, a negative swing is more likely to cause the register to store 0. The effectiveness of the injected fault is, therefore, determined by the polarity and the previously stored value in the register. Note that the EM affects all registers in the neighborhood and the attacker does not have precise control over this.

Laser fault injection. The target of the laser fault injection is the transistor layer of an IC. Through a focused laser beam, it produces electron hole pairs in the target area, which in turn might cause a high current drift, ultimately changing the output of a gate. Once the current drift collapses, the output switches back to its original value.

It has been shown that both memory and combinational gates are susceptible to laser fault injection [16,50] which can manifest in the effect of bit-level output flipping, outputting 1 or outputting 0, or changing the type of the combinational gate. Moreover, the target area of the laser fault injection ranges from a single gate to multiple (but limited) number of gates [6].

2.3 Modeling the Faults in the Circuit

In this section, we describe the terminology to model the injected faults. In ICs, various fault models can be used to describe the effects of the injected faults. We consider a set and reset faults that correspond to faulting a binary variable to get the logical values 1 and 0, respectively. A bit-flip fault corresponds to faulting a binary variable to get the complementary logical value.

In this work, we model the injected faults as manipulations at the gates excluding the wires, which includes set, reset, and bit-flip faults to any combinational logic or memory gates. Thus, faulting a gate is equivalent to altering its output. In principle, a faulty gate returns a faulty output for at least one input combination. We note that, in addition to the common combinational Boolean logic gates like AND, XOR, NOT, etc., digital circuits may also contain other types of combinational gates that are considered as part of the wire at an algorithmic level. One example of such gates is buffers, which are primarily used to regenerate the input. However, they can also be used to increase the propagation delay in the wire. While these buffers are not important at the functional algorithmic level, they become essential when modeling faults in the circuit since faults can be injected in them in the same way to the Boolean logic gates. *Therefore, we argue that such gates have to be part of the functional algorithmic level description of circuits when one considers fault adversaries.*

Beyond manipulating an injected fault at the gate level, a fault attack involves several additional factors in practice. We outline these factors in Section 3 by introducing the parameters that characterize the behavior of an adversary performing a fault injection.

3 Parameterized Fault Injection Adversary Model

In this section, we introduce our parameterized adversary model that encapsulates the control span a fault injection adversary can exert on an IC. This model encompasses the parameters such an adversary can actively control using a physical fault injection mechanism, namely: the number of fault injection events (n), fault location (l), fault timing (t), number of affected bits (b), duration of the injected fault (d), targeted type of gates (g), and fault type (p). Through these parameters, we can accurately represent an adversary by capturing the full control span of them on an IC in the event of a fault injection. We describe the parameters and summarize them in Table 1.

As noted in 1, the literature contains several studies that modeled a fault injection adversary using a range of parameters. We stress that the primary objective of our parameterized model is to precisely define a physical adversary. This allows us to question the alignment between analytical and mitigative adversary models, and the actual capabilities that an adversary can leverage through the physical fault injection mechanisms. For instance, Karaklajić *et al.* [31] characterized a fault adversary by encompassing the ability to control the fault location, time, effect, the number of affected bits, and the fault duration. However, this model overlooks some aspects such as the number of fault injection events and the targeted gate types. Likewise, Richter-Brockmann *et al.* [45] proposed a parameterized adversary model that captures an adversary through the number of affected bits, fault type, and fault location. Notably, this model is anticipated to have a better congruence with the models assumed in fault analysis methods. Thereby, it finds greater alignment with theoretical assessments rather than physical fault adversaries. This, in turn, prompts the central inquiry of this paper: To what degree do the analytical and mitigative adversary models align with practical scenarios? Given this context, our proposed parameterized adversary model emerges as a more holistic representation of a physical adversary with its comprehensive integration of parameters. Consequentially, our model stands as a bridge between analytical, mitigative and physical fault adversaries offering a framework to understand and counteract the fault injection vulnerabilities arising in practice.

Number of fault injection events (n)*.* This parameter defines the number of fault injection events an adversary performs during a specified time window (*e.g.,* the encryption/decryption operation, or a cycle). In particular, this parameter proves valuable, for instance, when describing an adversary injecting identical faults into replicated paths, or injecting faults at distinct cycles to circumvent some countermeasures. Note that we define the following parameters for each fault injection event.

Fault location (l)*.* This parameter defines the capabilities of an adversary over the location of an injected fault. Specifically, an adversary can have a *precise, loose*, or *no control* over the fault location. Having precise control implies that the adversary is able to inject a fault to a specific gate or cluster of a few gates, *i.e.,* can alter the specific bit(s). This level of control requires an adversary to have a high degree of knowledge of the implementation details. In contrast, having loose control implies that the adversary is able to target a specific (bigger) cluster of gates but has no/partial control over the location of the faulted bit(s). This level of control, being less precise, still requires some knowledge about the implementation. Lastly, having no control implies that the adversary is not able to target a specific gate, thereby precluding any direct control over the location of the faulted bit(s). Note that, the ability of an adversary to control the fault location highly depends on the fault injection setup. Hence, the model aims to capture various levels of control that an adversary possesses through different physical fault injection mechanisms.

Fault timing (t)*.* This parameter defines the capabilities of an adversary over the timing of the injected fault. Similar to fault location, an adversary can have a *precise, loose*, or *no control* over the fault timing. Having precise control over the timing implies that the adversary is able to inject a fault at a specific time (*i.e.,* in a specific clock cycle, an operation). Having loose control over timing implies that an adversary is able to target a set of operations or clock cycles. Lastly, no control implies that the adversary is not able to inject a fault at a specific time or period.

Number of affected bits (b). This parameter defines the number of bits affected by a fault injection. It is noteworthy that this parameter does not necessarily correspond to the number of observable faults in the circuit. That is, with time (*i.e.,* number of cycles) an injected fault might propagate to multiple locations as the erroneous value is subsequently used as input to other gates, or get ineffective. Moreover, a fault might have an ineffective effect on the target, *i.e.,* causing no change in the value.

Duration of the injected fault (d). This parameter defines the effectiveness period of an injected fault. The duration of an injected fault can be either *transient*, *persistent*, or *destructive*. A transient fault is effective for a limited period of time until the correct value is recovered again (*i.e.,* self-recoverable). This period varies depending on the time required to recover the original state and can be a fraction of a cycle, or multiple cycles. A persistent fault is effective as long as the fault injection finishes and the target variable is explicitly overwritten, implying duration of multiple cycles. A destructive fault damages the physical layer, *e.g.,* a fault in logic or memory that cannot be reversed, or the value of the target variable cannot be read anymore.

Targeted gate type (g). This parameter defines the type of the targeted gates in the circuit by a fault injection: combinational gates only, memory gates only, or both.

Fault type (p). This parameter defines the manifestation of the fault on the output(s) of the targeted gate(s). The fault type can be set, reset, flip, random, or custom. Set, reset, and flip faults refer to setting, resetting, and flipping the output of the targeted gate. Random fault refers to a fault that has an unpredictable outcome on the output of the targeted gate. Custom fault is used to define an adversary that is able to modify the mapping function of the targeted gate implements, which requires the strongest capabilities. Note that, in our parameterized adversary model, we deviate from the often used notation of *stuck-at 1/0* as also done by Richter-Brockmann *et al.* [45]. This is because stuck-at 0/1 faults are equivalent to reset/set faults for longer transient or persistent fault duration, and can thus be described by two fault parameters (*i.e.,* d and p).

In the next sections, we apply the parameterized fault adversary model to the physical fault injection mechanisms, and analytical and mitigative adversary models.

4 Parameterization of Physical Fault Injection Mechanisms

This section maps the capabilities of the physical fault injection mechanisms into the parameterized adversary model. All the described fault injection mechanisms impact the physical layer of the target device. However, they exhibit distinct characteristics leading to diverse fault scenarios in an IC. Clock/voltage glitching, for instance, affects the longest critical path, and depending on the

Table 1. The parameters defining the parameterized adversary model

Parameters	**Description**
Number of Fault Injection Events (n)	The number of physical fault injections performed in a specified time window
Fault Location (l)	**Precise:** Specific gate(s) **Loose:** Specific cluster of gates, no/partial control on which gates are affected **No control:** Random location
Fault Timing (t)	**Precise:** Specific clock cycle/operation **Loose:** Set of clock cycles/operations **No control:** Random timing
Number of Affected Bits (b)	The number of affected bits by the fault injection
Duration of the Injected Fault (d)	**Transient:** Limited, self-recoverable **Persistent:** Limited, needs to be explicitly overwritten **Destructive:** Irreversible
Targeted Gate Type (g)	**Combinational** gates only **Memory** gates only **Both**
Fault Type (p)	**Set:** Faulting to 1 **Reset:** Faulting to 0 **Random:** Random outcome **Flip:** Flipping the value **Custom:** Attacker specified gate modification

timing of the glitch (hence, the physical layout and the state of the circuit), more than one path might be affected. Therefore, while clock/voltage glitching does not require an expensive setup, it lacks precision in targeting a particular part of the IC. That is, a clock/voltage glitching adversary encounters constraints in terms of governing a precise fault location, in comparison to EM- and laser fault injection adversaries. On the other hand, EM-fault injection exhibits a higher spatial resolution compared to clock/voltage glitching, impacting all memory gates within the focus area of the setup. This distinction sets it apart from laser fault injection retaining the ability to selectively target individual gate(s).

Additionally, the number of gates faulted by clock/voltage glitching is random as it depends on the processed data and the underlying circuit at the targeted time. Nonetheless, if a circuit at the targeted time has a data path notably deeper than the others regarding the logic gates, the fault is inclined to occur within this data path, imposing a constraint on the number of gates affected by the fault. In contrast, EM- and laser fault injection exhibit heightened precision, as they allow for finer control over the specific target area on the IC. Notably, laser fault injection can target a fixed number of gates, thereby enhancing its precision beyond that of EM-fault injection. Another differentiating feature is that voltage glitching is incapable of performing multiple fault injection events within a single cycle (still can affect multiple bits). On the other hand, clock glitching, EM- and laser fault injection have the potential to perform multiple fault injection events within a single cycle. Additionally, the duration of fault injection can vary between these mechanisms. While EM- and laser fault injection can induce prolonged faults that last much longer than a single cycle [22], this is not the case for clock/voltage glitching which exhibit limitations in this regard. Moreover, with the exception of voltage glitching, all the aforementioned techniques can be executed within a fraction of a cycle.

We summarize the capabilities of the physical fault injection mechanisms in Table 2. Subsequently, in the next chapter, we parameterize the analytical adversary models of several fault/combined analysis methods discussed in the literature. Through this analysis, we illustrate the extent to which these analytical adversary models leverage the capabilities of the physical fault injection mechanisms.

Table 2. Physical fault injection mechanisms described as an adversary model

Fault Mechanism/ Parameters	Clock	Voltage	EM	Laser
(n) shots (per cycle)	Several	One	Several	Several
(l) location	Loose		Loose	Precise
(t) time	Precise		Precise	Precise
(b) bits	Random		Random	Several
(d) duration	Transient		Transient/ Destructive	Transient/ Persistent/ Destructive
(g) gates	Combinational		Memory	Both
(p) type	Random		Random	Custom

5 Parameterization of Analytical Adversary Models of Fault and Combined Analysis Methods

In this section, we revisit the analytical adversary models assumed in the most widely recognized fault and combined analysis methods in the literature. For each method, we map the analytical adversary model into our parameterized adversary model. We emphasize the necessity for a standardized adversary model by pointing out that the assumed models are not well-defined. A commonly agreed adversary model is crucial, not only for describing the fault adversaries of the fault/combined analysis methods but also for designing unified countermeasures against them. From this section on, we only consider the methods utilizing transient faults. Throughout the section, we denote the word length of the target implementations with w.

Note that, the literature often defines *the order of the fault attack* (t) as the total number of bits/variables altered during a cycle or the encryption/decryption operation, which is actually a function of parameters b and n. We will come back to the fault attack order and discuss it in Section 7.

Faults are often referred to as *effective* when the error propagates to the cipher output (*i.e.,* ciphertext is incorrect); or *ineffective* when the error propagation stops before reaching the cipher output (*i.e.,* ciphertext is correct). A method based on effective faults is DFA. However, two types of ineffective faults should be distinguished: faults that do not modify the intermediate value (*e.g.,* IFA), and faults that modify the intermediate value (*e.g.,* SIFA). It is easy to protect against IFA by using masking, while protection against SIFA is more challenging.

5.1 Fault Analysis Methods

In this section, we parameterize the analytical adversary models of the methods utilizing only fault injection mechanisms (versus combined analysis methods in Section 5.2), and list them in Table 3.

Differential Fault Analysis (DFA) DFA [8] exploits the differential information between correct and faulty ciphertexts obtained by injecting a fault to a state element during the last few rounds. Then, by analyzing the differential equations derived from both faulty and correct ciphertexts, it becomes possible to retrieve the last round key. Initially proposed on DES, DFA has also been applied to other algorithms such as AES [27]. We describe the analytical adversary model as follows:

(n) shots: One (l) location: Loose (t) time: Precise (b) bits: Up to w
(d) duration: Transient (g) gates: Both (p) type: Any

We note other methods that have the same exploit mechanism assuming the same analytical adversary model. For example, Algebraic Fault Attacks (AFA) [17] form algebraic equations and use an SAT solver afterward, Impossible Differential Fault Attacks (IDFA) [7] exploit the zero differentials rather than the high probability ones, and Linear Fault Analysis (LFA) [33] exploit the linear characteristics for some consecutive rounds.

Collision Fault Attack (CFA) CFA [9] combines the principles of DFA and collision attacks, using the collision information that is obtained when faulty and non-faulty encryptions have the same output. Then, the analysis of the collision information and the injected fault reveals information about the intermediate state. We describe the analytical adversary model as follows:

(n) shots: One (l) location: Precise (t) time: Precise (b) bits: One
(d) duration: Transient (g) gates: Memory (p) type: Flip

Fault Sensitivity Analysis (FSA) FSA [32] observes the data dependency of the fault occurrence as the intensity of the fault injection mechanism increases. The intensity of the fault injection mechanism could be controlled through adjustments in power supply reduction or clock period elongation. FSA assumes that the attacker begins fault injection at an intensity level that results in the correct ciphertext. They gradually increase the intensity until the fault injection has a nonzero success rate, and eventually, a success rate of one. The attacker uses this fault sensitivity information to recover secret information as it depends on the secret data. The analytical adversary model is described as follows:

(n) shots: One (l) location: Loose (t) time: Precise (b) bits: Up to w
(d) duration: Transient (g) gates: Comb. (p) type: Random

We note the extension of FSA, Collision FSA [35], that extends FSA with correlation enhanced collision side-channel attacks, and Differential Fault Intensity Analysis (DFIA) [26] that uses fault intensity and faulty output in the statistical analysis, assume the same analytical adversary model.

Safe Error Attack (SEA), Ineffective Fault Analysis (IFA) SEA [57] was initially proposed for RSA targeting the right-to-left exponentiation, but has been shown to be applicable to other algorithms. Essentially, SEA exploits *safe errors* that do not alter the output revealing information about the path executed by the algorithm, thereby revealing some secret information. In this context, IFA [14] applied to symmetric key algorithms shares a common approach with SEA by not altering the output. Whereas SEA reveals algorithm specific information by actually modifying the intermediate values, IFA reveals information about the targeted variable by not modifying the intermediate value. That is, if an attacker receives a correct output, it indicates that the injected fault did not modify the targeted variable. Here, the attacker needs to know the type of the injected fault as it reveals the value of the faulted variable. We describe the analytical adversary model assumed in IFA as follows:

(n) shots: One (l) location: Precise (t) time: Precise (b) bits: Up to w
(d) duration: Transient (g) gates: Both (p) type: Set, reset, custom

Given the parameters of the analytical adversary model, the attack can be carried out by a laser fault injection as the method calls for a strong adversary which can inject a known fault.

Statistical Fault Attacks (SFA) SFA [25] was originally proposed for AES introducing a bias to an intermediate variable through fault injection. In essence, due to the introduced bias, the statistical distribution of the targeted variable obtained from the faulty ciphertexts is non-uniform, which can be exploited by the attacker to perform key recovery.

SFA is performed via clock glitching and laser fault injection by Dobraunig *et al.* [20]. However, it is also possible to carry out the attack via EM-fault injection or voltage glitching as the analysis method does not call for strong assumptions on the fault. The analytical adversary model can be described as follows:

(n) shots: One	(l) location: Loose	(t) time: Precise	(b) bits: Up to w
(d) duration: Transient	(g) gates: Both	(p) type: Any	

Statistical Ineffective Fault Attacks (SIFA) Similar to SFA, SIFA [21,19] also exploits the bias introduced to the target variable by the fault injection. However, SIFA analyses the statistical distribution of the targeted variable obtained from the correct ciphertexts.

We categorize SIFA in two: SIFA-1 [21] and SIFA-2 [19] as in [49]. SIFA-1 assumes a fault is injected to a state variable, or to a linear operation. On the other hand, SIFA-2 assumes a fault is injected to non-linear operations like an S-box. SIFA-2 stands as a more powerful method as masking with detection countermeasures do not protect against it, whereas they protect against SIFA-1. All fault types except bit-flip and random faults can result in SIFA-1. On the contrary, SIFA-2 can only be performed via a bit-flip and a random fault. The attack is performed via clock/voltage glitches, however, it is possible to carry out the attack via EM and laser fault injections. The analytical adversary model can be described as follows:

(n) shots: One	(l) location: Loose	(t) time: Precise	(b) bits: Up to w
(d) duration: Transient	(g) gates: Both	(p) type: SIFA-1 - Set, reset, custom SIFA-2 - Bit flip, random	

We note that Fault Intensity Map Analysis (FIMA) [40] generalizes FSA, DFIA, and SIFA by employing biased fault injections with varying intensities. FIMA assumes the same analytical adversary model as SIFA-1 and -2.

Fault Template Attacks (FTA) FTA [48] exploits the dependency of the fault activation and propagation on the secret data. Although the analysis is similar to SIFA, FTA does not require the correct/faulty outputs, but only the knowledge of the output being faulty or not. Moreover, while SIFA is demonstrated only in the last rounds, FTA extends the analysis to the middle rounds. FTA builds a fault pattern for different fault locations collected from different cipher executions depending on whether the fault is effective or not, which happens at the offline phase to characterize the circuit. Then, in the online phase, the templates are matched to the execution that is being analyzed.

The authors perform the attack via EM-fault injection assuming the following analytical adversary model:

(n) shots: One	(l) location: Precise	(t) time: Precise	(b) bits: One
(d) duration: Transient	(g) gates: Both	(p) type: Set, reset, bit flip	

Fault Correlation Analysis (FCA) FCA [55] investigates the relation between side-channel analysis and fault injection. The probability of a fault occurring is dependent on the data being processed, and the operation being performed, thereby, it is hypothesized to be correlated to the power consumption. The main idea of FCA is to turn the observed faults into a probability at a given time and to repeat this at different points in time to get probability traces, which are equivalent to power traces. These traces are then exploited with a standard side-channel analysis. The analytical adversary model can be described as follows:

(n) shots: One	(l) location: Loose	(t) time: Precise	(b) bits: Up to w
(d) duration: Transient	(g) gates: Both	(p) type: Random	

Statistical Effective Fault Attacks (SEFA) Similar to SIFA, SEFA [56] exploits the non-uniformity of the distribution of an intermediate value. While SIFA utilizes ineffective ciphertexts, SEFA utilizes non-faulty ciphertexts corresponding to effective faults. Thus, SEFA requires less number of ciphertexts to do a key-recovery attack. In general, SEFA exhibits better performance than SIFA in the presence of fault injection setup noise.

Similar to SIFA, the attack is performed via clock/voltage glitches by the authors using the same analytical adversarial model, described as follows:

(n) shots: One	(l) location: Loose	(t) time: Precise	(b) bits: Up to w
(d) duration: Transient	(g) gates: Both	(p) type: Any	

5.2 Combined Analysis Methods

In this section, we parameterize the analytical adversary models of the methods utilizing both fault injection and side-channels in a combined setting, and list them in Table 3.

Passive and Active Combined Attacks (PACA) PACA [3], originally proposed for RSA, combines passive and active analysis. It exploits the fault countermeasures reacting at the end of the execution by recovering the secret via classical power analysis before the countermeasure takes effect. Clavier *et al.* [15] applied this analysis concept to a masked AES implementation, which we consider in this section. The analysis assumes a fault that sets the output of an XOR operation to zero (or a constant value) which is injected to the first key addition before the first round. Then, using the differentials obtained from correct and faulty ciphertexts, and the power curves of the random values used in masking, the attacker performs a key recovery. We describe the analytical adversary model as follows:

(n) shots: One	(l) location: Precise	(t) time: Precise	(b) bits: Up to w
(d) duration: Transient	(g) gates: Comb.	(p) type: Set, reset, custom	

A Combined Analysis on a Protected AES This analysis [46] targets a fault analysis resistant and masked AES implementation by combining DFA and Correlation Power Analysis (CPA) [12]. The idea is to utilize fault injection to affect the last but one round of the key scheduling algorithm to fault the last two round keys. However, as the faulty ciphertexts are being suppressed due to fault detection/correction, side-channel information is instead used to collect the corresponding information for these faulty ciphertexts. Then, the analysis follows the round key retrieving strategy of DFA, through the differential equations. We describe the analytical adversary model as follows:

(n) shots: One (l) location: Loose (t) time: Precise (b) bits: Up to w
(d) duration: Transient (g) gates: Both (p) type: Any

SCA-Enhanced Fault Template Attacks (SCA-FTA) SCA-FTA [47] enhances FTA using side-channel leakage in the presence of faults, and building the templates using the leakage information from the detection and correction operations. SCA-FTA exploits the observations of the S-box output differentials in the presence of faults that leak information about the S-box inputs. The analysis works similarly to FTA. However, it uses the side-channel leakage from the error-handling logic to build the templates rather than the knowledge of the effectiveness of the fault. The analysis assumes the same analytical adversary model used in FTA:

(n) shots: One (l) location: Precise (t) time: Precise (b) bits: One
(d) duration: Transient (g) gates: Both (p) type: Set, reset, bit flip

Table 3. Mapping of the adversary models of the presented fault/combined attacks where S, R, BF, C and RM refer to set, reset, bit flip, custom and random, respectively.

Parameters/	(n)	(l)	(t)	(b)	(d)	(g)	(p)
Attacks	shots	location	time	bits	duration	gates	type
DFA [8]	One	Loose	Precise	Up to w	Transient	Both	Any
CFA [9]	One	Precise	Precise	One	Transient	Mem.	BF
FSA [32]	One	Loose	Precise	Up to w	Transient	Comb.	Random
IFA [14]	One	Precise	Precise	Up to w	Transient	Both	S,R,C
SFA [25]	One	Loose	Precise	Up to w	Transient	Both	Any
SIFA1 [21] SIFA2 [19]	One	Loose	Precise	Up to w	Transient	Both	S,R,C BF,RM
FTA [48]	One	Precise	Precise	One	Transient	Both	S,R,BF
FCA [55]	One	Loose	Precise	Up to w	Transient	Both	RM
SEFA [56]	One	Loose	Precise	Up to w	Transient	Both	Any
PACA [3]	One	Precise	Precise	Up to w	Transient	Comb.	S,R,C
Roche *et al.* [46]	One	Loose	Precise	Up to w	Transient	Both	Any
SCA-FTA [47]	One	Precise	Precise	One	Transient	Both	S,R,BF

6 Parameterization of Mitigative Adversaries Assumed in Countermeasures

In this section, we revisit the mitigative adversary models assumed in several countermeasures. To provide a comprehensive evaluation, we map the mitigative adversary models used in each countermeasure into our parameterized adversary model. We stress that the mitigative adversary models assumed in these countermeasures are not always precisely defined, which is partially due to the lack of a standardized adversary model. Furthermore, many of these countermeasures are designed to protect against specific fault analysis methods, rather than physical fault injection mechanisms that an adversary may utilize. This makes it more challenging to provide complete protection against all known analysis methods as each method may need to be addressed individually. We list a summary of the parameters used to describe the mitigative adversary models assumed in the countermeasures in Table 4.

ParTI ParTI [51] assumes an adversary possessing both SCA and faulting capabilities. Its design predates the introduction of SIFA, and at the time it was designed, it was secure against all known fault attacks. However, despite not being explicitly designed to protect against SIFA, ParTI offers protection against SIFA-1-like attacks. It employs threshold implementations (TI) combined with error detection using linear codes. More specifically, ParTI makes use of a systematic code in which the prediction functions are also masked to secure against SCA and all the listed fault attacks exploiting effective faults and ineffective faults with the exception of SIFA-2-like attacks. We describe the mitigative adversary model assumed by the authors using the parameters as follows:

(n) shots: Up to k (l) location: Any (t) time: Any (b) bits: Up to t
(d) duration: Transient (g) gates: Both (p) type: Any

We note that the countermeasure proposed by Richter-Brockmann *et al.* [44] extends the approach combining TI and linear codes by dynamically changing the applied (non-systematic) linear codes as a hiding technique, offering higher-order side-channel security. Taking a different approach, RS-Mask [41] extends TI with random space masking.

CAPA CAPA [43] provides provable security against higher-order SCA, higher-order fault attacks, and combined attacks by leveraging the principles of the MPC protocol SPDZ. Unlike the common SCA and analytical adversary models that assume the t-probing model [30], and faulting up to a limited number of gates, CAPA adopts a unique approach in its mitigative adversary model: *The Tile Probe and Fault Model.* This model assumes that the chip is partitioned into tiles connected by wires having their own combinational and control logic, and PRNGs. Additionally, each tile processes at most one share of an intermediate variable. Unlike the standard models, the Tile Probe and Fault Model allows an attacker to probe t tiles (out of $t + 1$ tiles) with all their possessed intermediate values, making it more robust than the t-probing wire model. Similarly, the model allows an attacker to inject a random fault to any variable possessed by any of the tiles. It also allows an attacker to inject a non-stochastic fault to any variable possessed by up to t tiles. The first type of faults can be injected using clock glitches while the second type requires a laser fault injection.

Despite being designed prior to the introduction of SIFA, CAPA provides comprehensive security against all the listed effective and ineffective fault attacks, including SIFA-2. It is worth noting that at the time of SIFA publication, it was the only provable secure countermeasure that existed and was secure against SIFA-2. We formulate the mitigative adversary model assumed by the authors (*i.e.,* the Tile Probe and Fault Model) using the parameters as follows:

(n) shots: Stochastic any, else up to k (l) location: Any (t) time: Any
(b) bits: Stochastic any, else up to t
(d) duration: Transient (g) gates: Both (p) type: Any

M&M M&M [34] protects against fault attacks by ensuring data integrity using information-theoretic MAC tags extending any SCA-secure masking scheme. The design of M&M was inspired by the principles of CAPA. However, unlike CAPA, M&M assumes a simplified mitigative adversary model that operates on wires and gates rather than tiles, while still distinguishing between the two types of adversaries. Besides providing security against SCA due to the underlying masking scheme, M&M provides generic order security against all the listed attacks, explicitly excluding SIFA-2-like attacks. M&M infects the output if a fault is detected. We describe the mitigative adversary model using the parameters as follows:

(n) shots: Stochastic any, else up to k (l) location: Any (t) time: Any
(b) bits: Stochastic any, else up to t
(d) duration: Transient (g) gates: Both (p) type: Any

We note that Hirata *et al.* [29] extends M&M to resist certain specific SIFA-2 attacks caused by clock glitches. Unlike M&M, it employs a detection mechanism instead of infection.

Transform-and-Encode (TaE) TaE [49] was designed based on two strategies, namely *transform* and *encode*. The transform strategy aims to randomize the state such that injected faults at the state do not cause biased distributions. This strategy particularly protects against SIFA-1, where masking is a potential candidate. Therefore, it can be implemented using any SCA secure masking scheme, providing protection against both SCA and SIFA-1-like attacks. The encode strategy utilizes error correction techniques to protect against SIFA-2-like attacks. In this manner, TaE provides protection against all the listed fault attacks utilizing effective and ineffective faults. We describe the mitigative adversary model using the parameters as follows:

(n) shots: Up to k (l) location: Any (t) time: Any (b) bits: Up to t
(d) duration: Transient (g) gates: Both (p) type: Any

We note that DOMREP [28] uses a similar approach as TaE combining domain-oriented masking and repetition codes, and the countermeasure by Breier *et al.* [11] uses error correction codes at gate level.

Impeccable Circuits (ImC) I, II, III ImC schemes are based on linear codes: ImC I [1] utilizes error detection, ImC II [52] utilizes error correction, and ImC III [42] utilizes both error detection and correction. To handle fault propagation, the authors proposed using additional error check/correction points and forced independence. The forced independence property requires that no gate is shared between any two component circuits, where each component circuit computes a single output bit. However, these properties come with increased area overhead.

ImC I was specifically designed to secure against effective faults. ImC II utilizes error correction, which in turn protects against both effective and ineffective faults. The authors report that ImC II has no significant performance benefits when compared to majority voting, which led the authors to design ImC III combining error detection and correction. Specifically, ImC III corrects faults as long as the number of faulty bits is below a threshold, otherwise, it detects the fault if the number of faulty bits is again below another threshold depending on the used linear code.

ImC schemes are not SCA secure by their nature, however, hardware Boolean masking schemes can be easily implemented as the linear codes do not increase the algebraic degree of the construction. ImC I, II, and III share the common mitigative adversary model that allows to fault up to t bits in a *single* clock cycle of the entire operation (*i.e.,* a univariate adversary model), or at *multiple* clock cycles (*i.e.,* a multivariate adversary model). We describe the model as follows:

(n) shots: Up to k	(l) location: Any	(t) time: Any	(b) bits: Up to t
(d) duration: Transient	(g) gates: Both	(p) type: Any	

Permutations and Fine-Grained Fault Detection Daemen *et al.* [18] proposed two strategies aimed at thwarting SIFA-1 and -2. The first technique is to use permutations as the building blocks. The second technique is to use a fine-grained fault detection mechanism that can detect faults before they become ineffective later in the circuit.

The authors have a slightly different approach to describe their mitigative adversary model. Injected faults are abstracted at the basic circuit level (*i.e.,* non-complete permutations) which do not depend on any secrets as the basic circuits are non-complete. Then, a single fault is defined as faulting a single basic circuit, which modifies the circuit such that it returns an incorrect output for at least one input combination. We describe the mitigative adversary model using the parameters as follows:

(n) shots: One	(l) location: Any	(t) time: Any	(b) bits: Up to t
(d) duration: Transient	(g) gates: Both	(p) type: Any	

We note that FRIET [53], a duplex-based authenticated encryption scheme, provides first order SIFA protection using the countermeasures introduced in [18].

Combined Private Circuits (CPC) Combined Private Circuits [23] applies the core ideas behind Probe-Isolating Non-Interference (PINI) [13] to both fault and combined security. The authors propose an attack against CINI-MINIS [24], and new (fixed) composable gadgets. The proposed gadgets rely on both masking and spacial replication (*i.e.,* error correction via majority voting). We describe the mitigative adversary model as follows:

(n) shots: Up to k	(l) location: Any	(t) time: Any	(b) bits: Up to t
(d) duration: Transient	(g) gates: Both	(p) type: Any	

Table 4. Mapping the adversary models of the presented countermeasures to the parameterized model

Parameters/ Attacks	(n) shots	(l) location	(t) time	(b) bits	(d) duration	(g) gates	(p) type
ParTI [51]	Up to k	Any	Any	Up to t	Transient	Both	Any
CAPA [43]	Any RM Up to k	Any	Any	Any RM Up to t	Transient	Both	Any
M&M [34]	Any RM Up to k	Any	Any	Any RM Up to t	Transient	Both	Any
TaE [49]	Up to k	Any	Any	Up to t	Transient	Both	Any
ImC [1,52,42]	Up to k	Any	Any	Up to t	Transient	Both	Any
Permutations [18]	One	Any	Any	Up to t	Transient	Both	Any
CPC [23]	Up to k	Any	Any	Up to t	Transient	Both	Any

7 Discussion

Our work presents a parameterized adversary model into which we mapped the physical adversary models reflecting the capabilities of physical fault injection mechanisms, and the existing analytical and mitigative adversary models. Through these three mappings, our parameterized adversary model facilitates a comprehensive evaluation of the extent to which analytical and mitigative adversary models correspond to real-world scenarios.

We start our analysis with the following findings, based on Table 2. Upon mapping the physical fault injection mechanisms into the parameterized adversary model, it becomes evident that these mechanisms exhibit a notable degree of precision, either in terms of time or both time and location. Moreover, this mapping highlights their considerable power, enabling attackers to inject as many faults as desired. In light of these features, we can categorize these mechanisms into two groups: (i) high precision with relatively small target areas, and (ii) low precision with relatively large target areas, or more precisely:

i) The first group of physical fault injection mechanisms empowers attackers with the capacity to precisely target specific gates with the desired fault types. However, their target location on ASIC is confined to a few gates, and once the location is selected at the beginning of the encryption, it remains fixed. Despite this limitation, the attacker can still perform several fault injection events within a cycle, and keep the injection active over several cycles. Laser fault injection is an example of such an injection mechanism.

ii) The second group of physical fault injection mechanisms, while lacking such precision, targets larger areas, affecting more adjacent gates than those originally intended. Although such an attacker can simultaneously affect multiple gates, they have limited control over the resulting faulty values. Additionally, similar to the first group, the target location of the mechanism is static once chosen at the beginning of the encryption. Nonetheless, the attacker is capable of performing multiple fault injection events within a cycle, and keeping the injection active over several cycles. Clock and voltage glitches, as well as EM-fault injection, exemplify such injection mechanisms possessing these features.

We note that this categorization also matches well with the different fault types (p) of the methods, namely the second group can introduce only random faults to the intermediate value, while the first group can introduce all possible fault types. Both groups share the common characteristic of being capable of having only a few fixed target locations (non-adaptively), since too many lasers or EM-probes cannot simultaneously inject faults. Most importantly, both groups have the capability to inject faults as many times as desired and thus fault as many bits as desired.

In summary, *two types of adversaries can be distinguished: the first one injects only a few (upper bounded) but precise faults; whereas the second one injects many (unlimited) but random faults.*

However, as Table 3 indicates, fault and combined analysis methods do not fully utilize the capabilities of physical fault injection mechanisms, demanding only a fraction of them for a successful analysis. Specifically, these methods exploit a single injection over the entire encryption process, only when the limited number of bits have been faulted. To the best of our knowledge, there have been no proposed fault analysis methods requiring multiple fault injections (for ASIC implementations).

Table 4 shows that the mitigative adversary models tend to align better with the analytical adversary models rather than the capabilities of the physical fault injection mechanisms. The classical analytical adversary model is assuming *precise but a limited number of faults, i.e., bounded order of attack. In other words, it is assumed that an attacker can fault only a limited number (up to t) of bits/variables within a cycle or during the encryption process, and that they can always introduce precise faults. However, two exceptions to this trend are CAPA and M&M, which consider also attackers injecting many but random faults.*

We note that the mitigative adversary models' assumption that the order of attack is bounded is not always correct, as we have shown the physical injection mechanisms exhibit no such limitations. Moreover, whenever the attacker can introduce an unbounded number of faults they are no longer capable of being precise on the type of the faults. *Due to this discrepancy between the classical analytical adversary models and the physical reality, proposed countermeasures may provide only limited protection against physical fault injection mechanisms, despite their provable security within a more restricted mitigative adversary model.*

Conversely, the mitigative adversary models allow the attacker to target up to l locations and sometimes to be adaptive, while practical scenarios limit the injection to a few fixed positions. As such, the countermeasures may be considered *over-designed* with respect to the actual capabilities of the physical fault injection mechanisms.

This discussion leads us to the conclusion that in contrast to side-channel attacks, fault attacks do not have a known limitation regarding the number of fault injection events as well as the number of bits being faulted due to the capabilities of the physical fault injection mechanisms. SCA is known to be constrained by the noise level in the power/EM traces, which limits the order of the attack. However, for fault attacks, an attacker can inject several faults in a single clock cycle, potentially targeting a few locations based on the specific implementation and the fault setup, and hence the attacker can go beyond the order of attack chosen by the countermeasure.

We finish this overview by posing several open questions. We strongly believe that a more comprehensive and unified fault/combined adversary model must be established. The parameterized adversarial model presented in this work represents the first step towards such a model. We suggested two such sub-models, noting that more characteristics for them can be specified. The next step would aim to design improved countermeasures that are provably secure in this unified model. The error-correction and error-detection mechanisms used in countermeasures are typically limited in their capacity to handle a large number of faults. Thus, a mechanism is required that can provide finer granularity before the errors accumulate to an excessive extent. However, it remains an open question whether such a mechanism is achievable even if the fault propagation is inherently limited by design and given the capabilities of the fault injection mechanisms which can fault multiple bits at a single location. In addition, since all fault injection mechanisms have precise timing and duration control, time redundancy as a countermeasure seems to be more vulnerable than spatial redundancy. Probing the error propagation framework [23] matches well the classical mitigative adversary. However, when the number of faults is unbounded and they can happen on "any" location/value injecting a random value, the investigation of the propagations might become infeasible. A modification or extension of such a framework will be required. All those open questions we leave as future work.

Acknowledgements This work was supported by CyberSecurity Research Flanders with reference number VR20192203.

References

1. Aghaie, A., Moradi, A., Rasoolzadeh, S., Shahmirzadi, A.R., Schellenberg, F., Schneider, T.: Impeccable circuits. IEEE Trans. Computers **69**(3), 361–

376 (2020). https://doi.org/10.1109/TC.2019.2948617, https://doi.org/10.1109/TC.2019.2948617

2. Agoyan, M., Dutertre, J., Naccache, D., Robisson, B., Tria, A.: When clocks fail: On critical paths and clock faults. In: Gollmann, D., Lanet, J., Iguchi-Cartigny, J. (eds.) Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, Passau, Germany, April 14-16, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6035, pp. 182–193. Springer (2010). https://doi.org/10.1007/978-3-642-12510-2_13, https://doi.org/10.1007/978-3-642-12510-2_13
3. Amiel, F., Villegas, K., Feix, B., Marcel, L.: Passive and active combined attacks: Combining fault attacks and side channel analysis. In: Breveglieri, L., Gueron, S., Koren, I., Naccache, D., Seifert, J. (eds.) Fourth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2007, FDTC 2007: Vienna, Austria, 10 September 2007. pp. 92–102. IEEE Computer Society (2007). https://doi.org/10.1109/FDTC.2007.4318989, https://doi.org/10.1109/FDTC.2007.4318989
4. Aumüller, C., Bier, P., Fischer, W., Hofreiter, P., Seifert, J.: Fault attacks on RSA with CRT: concrete results and practical countermeasures. In: Jr., B.S.K., Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2523, pp. 260–275. Springer (2002). https://doi.org/10.1007/3-540-36400-5_20, https://doi.org/10.1007/3-540-36400-5_20
5. Bar-El, H., Choukri, H., Naccache, D., Tunstall, M., Whelan, C.: The sorcerer's apprentice guide to fault attacks. Proc. IEEE **94**(2), 370–382 (2006). https://doi.org/10.1109/JPROC.2005.862424, https://doi.org/10.1109/JPROC.2005.862424
6. Bartkewitz, T., Bettendorf, S., Moos, T., Moradi, A., Schellenberg, F.: Beware of insufficient redundancy an experimental evaluation of code-based FI countermeasures. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2022**(3), 438–462 (2022). https://doi.org/10.46586/tches.v2022.i3.438-462, https://doi.org/10.46586/tches.v2022.i3.438-462
7. Biham, E., Granboulan, L., Nguyen, P.Q.: Impossible fault analysis of RC4 and differential fault analysis of RC4. In: Gilbert, H., Handschuh, H. (eds.) Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3557, pp. 359–367. Springer (2005). https://doi.org/10.1007/11502760_24, https://doi.org/10.1007/11502760_24
8. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Jr., B.S.K. (ed.) Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1294, pp. 513–525. Springer (1997). https://doi.org/10.1007/BFb0052259, https://doi.org/10.1007/BFb0052259
9. Blömer, J., Krummel, V.: Fault based collision attacks on aes. In: Breveglieri, L., Koren, I., Naccache, D., Seifert, J.P. (eds.) Fault Diagnosis and Tolerance in Cryptography. pp. 106–120. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
10. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults (extended abstract). In: Fumy, W. (ed.) Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-

15, 1997, Proceeding. Lecture Notes in Computer Science, vol. 1233, pp. 37–51. Springer (1997). https://doi.org/10.1007/3-540-69053-0_4, https://doi.org/10.1007/3-540-69053-0_4

11. Breier, J., Khairallah, M., Hou, X., Liu, Y.: A countermeasure against statistical ineffective fault analysis. IEEE Trans. Circuits Syst. **67-II**(12), 3322–3326 (2020). https://doi.org/10.1109/TCSII.2020.2989184, https://doi.org/10.1109/TCSII.2020.2989184
12. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings. Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer (2004). https://doi.org/10.1007/978-3-540-28632-5_2, https://doi.org/10.1007/978-3-540-28632-5_2
13. Cassiers, G., Standaert, F.: Trivially and efficiently composing masked gadgets with probe isolating non-interference. IEEE Trans. Inf. Forensics Secur. **15**, 2542–2555 (2020). https://doi.org/10.1109/TIFS.2020.2971153, https://doi.org/10.1109/TIFS.2020.2971153
14. Clavier, C.: Secret external encodings do not prevent transient fault analysis. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4727, pp. 181–194. Springer (2007). https://doi.org/10.1007/978-3-540-74735-2_13, https://doi.org/10.1007/978-3-540-74735-2_13
15. Clavier, C., Feix, B., Gagnerot, G., Roussellet, M.: Passive and active combined attacks on aes combining fault attacks and side channel analysis. In: 2010 Workshop on Fault Diagnosis and Tolerance in Cryptography. pp. 10–19 (2010). https://doi.org/10.1109/FDTC.2010.17
16. Courbon, F., Loubet-Moundi, P., Fournier, J.J.A., Tria, A.: Adjusting laser injections for fully controlled faults. In: Prouff, E. (ed.) Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8622, pp. 229–242. Springer (2014). https://doi.org/10.1007/978-3-319-10175-0_16, https://doi.org/10.1007/978-3-319-10175-0_16
17. Courtois, N.T., Ware, D., Jackson, K.M.: Fault-algebraic attacks on inner rounds of des. In: The eSmart 2010 European Smart Card Security Conference (2010)
18. Daemen, J., Dobraunig, C., Eichlseder, M., Groß, H., Mendel, F., Primas, R.: Protecting against statistical ineffective fault attacks. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2020**(3), 508–543 (2020). https://doi.org/10.13154/tches.v2020.i3.508-543, https://doi.org/10.13154/tches.v2020.i3.508-543
19. Dobraunig, C., Eichlseder, M., Groß, H., Mangard, S., Mendel, F., Primas, R.: Statistical ineffective fault attacks on masked AES with fault countermeasures. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11273, pp. 315–342. Springer (2018). https://doi.org/10.1007/978-3-030-03329-3_11, https://doi.org/10.1007/978-3-030-03329-3_11
20. Dobraunig, C., Eichlseder, M., Korak, T., Lomné, V., Mendel, F.: Statistical fault attacks on nonce-based authenticated encryption schemes. In: Cheon, J.H., Tak-

agi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 369–395 (2016). https://doi.org/10.1007/978-3-662-53887-6_14, https://doi.org/10.1007/978-3-662-53887-6_14

21. Dobraunig, C., Eichlseder, M., Korak, T., Mangard, S., Mendel, F., Primas, R.: Sifa: Exploiting ineffective fault inductions on symmetric cryptography. Transactions on Cryptographic Hardware and Embedded Systems **2018**, 547–572 (11 2018). https://doi.org/10.13154/tches.v2018.i3.547-572
22. Dumont, M., Lisart, M., Maurine, P.: Electromagnetic fault injection : How faults occur. In: 2019 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2019, Atlanta, GA, USA, August 24, 2019. pp. 9–16. IEEE (2019). https://doi.org/10.1109/FDTC.2019.00010, https://doi.org/10.1109/FDTC.2019.00010
23. Feldtkeller, J., Güneysu, T., Moos, T., Richter-Brockmann, J., Saha, S., Sasdrich, P., Standaert, F.X.: Combined private circuits - combined security refurbished. p. 1341 (2023), https://eprint.iacr.org/2023/1341
24. Feldtkeller, J., Richter-Brockmann, J., Sasdrich, P., Güneysu, T.: CINI MINIS: domain isolation for fault and combined security. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022. pp. 1023–1036. ACM (2022). https://doi.org/10.1145/3548606.3560614, https://doi.org/10.1145/3548606.3560614
25. Fuhr, T., Jaulmes, É., Lomné, V., Thillard, A.: Fault attacks on AES with faulty ciphertexts only. In: Fischer, W., Schmidt, J. (eds.) 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013. pp. 108–118. IEEE Computer Society (2013). https://doi.org/10.1109/FDTC.2013.18, https://doi.org/10.1109/FDTC.2013.18
26. Ghalaty, N.F., Yuce, B., Taha, M.M.I., Schaumont, P.: Differential fault intensity analysis. In: Tria, A., Choi, D. (eds.) 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014, Busan, South Korea, September 23, 2014. pp. 49–58. IEEE Computer Society (2014). https://doi.org/10.1109/FDTC.2014.15, https://doi.org/10.1109/FDTC.2014.15
27. Giraud, C.: Dfa on aes. In: Dobbertin, H., Rijmen, V., Sowa, A. (eds.) Advanced Encryption Standard – AES. pp. 27–41. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
28. Gruber, M., Probst, M., Karl, P., Schamberger, T., Tebelmann, L., Tempelmeier, M., Sigl, G.: Domrep-an orthogonal countermeasure for arbitrary order side-channel and fault attack protection. IEEE Trans. Inf. Forensics Secur. **16**, 4321–4335 (2021). https://doi.org/10.1109/TIFS.2021.3089875, https://doi.org/10.1109/TIFS.2021.3089875
29. Hirata, H., Miyahara, D., Arribas, V., Li, Y., Miura, N., Nikova, S., Sakiyama, K.: All you need is fault: Zero-value attacks on AES and a new λ-detection m&m. IACR Cryptol. ePrint Arch. p. 1129 (2023), https://eprint.iacr.org/2023/1129
30. Ishai, Y., Sahai, A., Wagner, D.A.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2729, pp. 463–481. Springer (2003). https://doi.org/10.1007/978-3-540-45146-4_27, https://doi.org/10.1007/978-3-540-45146-4_27

31. Karaklajic, D., Schmidt, J., Verbauwhede, I.: Hardware designer's guide to fault attacks. IEEE Trans. Very Large Scale Integr. Syst. **21**(12), 2295–2306 (2013). https://doi.org/10.1109/TVLSI.2012.2231707, https://doi.org/10.1109/TVLSI.2012.2231707
32. Li, Y., Sakiyama, K., Gomisawa, S., Fukunaga, T., Takahashi, J., Ohta, K.: Fault sensitivity analysis. In: Mangard, S., Standaert, F.X. (eds.) Cryptographic Hardware and Embedded Systems, CHES 2010. pp. 320–334. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
33. Liu, Z., Gu, D., Liu, Y., Li, W.: Linear fault analysis of block ciphers. In: Bao, F., Samarati, P., Zhou, J. (eds.) Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore, June 26-29, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7341, pp. 241–256. Springer (2012). https://doi.org/10.1007/978-3-642-31284-7_15, https://doi.org/10.1007/978-3-642-31284-7_15
34. Meyer, L.D., Arribas, V., Nikova, S., Nikov, V., Rijmen, V.: M&m: Masks and macs against physical attacks. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(1), 25–50 (2019). https://doi.org/10.13154/tches.v2019.i1.25-50, https://doi.org/10.13154/tches.v2019.i1.25-50
35. Moradi, A., Mischke, O., Paar, C., Li, Y., Ohta, K., Sakiyama, K.: On the power of fault sensitivity analysis and collision side-channel attacks in a combined setting. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6917, pp. 292–311. Springer (2011). https://doi.org/10.1007/978-3-642-23951-9_20, https://doi.org/10.1007/978-3-642-23951-9_20
36. Ning, B., Liu, Q.: Modeling and efficiency analysis of clock glitch fault injection attack. In: Asian Hardware Oriented Security and Trust Symposium, AsianHOST 2018, Hong Kong, China, December 17-18, 2018. pp. 13–18. IEEE (2018). https://doi.org/10.1109/AsianHOST.2018.8607175, https://doi.org/10.1109/AsianHOST.2018.8607175
37. Ordas, S., Guillaume-Sage, L., Maurine, P.: EM injection: Fault model and locality. In: Homma, N., Lomné, V. (eds.) 2015 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2015, Saint Malo, France, September 13, 2015. pp. 3–13. IEEE Computer Society (2015). https://doi.org/10.1109/FDTC.2015.9, https://doi.org/10.1109/FDTC.2015.9
38. Ordas, S., Guillaume-Sage, L., Maurine, P.: Electromagnetic fault injection: the curse of flip-flops. J. Cryptogr. Eng. **7**(3), 183–197 (2017). https://doi.org/10.1007/s13389-016-0128-3, https://doi.org/10.1007/s13389-016-0128-3
39. Quisquater, J.J., Samyde, D.: Eddy current for magnetic analysis with active sensor. In: Proceedings of ESmart 2002 (2002)
40. Ramezanpour, K., Ampadu, P., Diehl, W.: FIMA: fault intensity map analysis. In: Polian, I., Stöttinger, M. (eds.) Constructive Side-Channel Analysis and Secure Design - 10th International Workshop, COSADE 2019, Darmstadt, Germany, April 3-5, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11421, pp. 63–79. Springer (2019). https://doi.org/10.1007/978-3-030-16350-1_5, https://doi.org/10.1007/978-3-030-16350-1_5
41. Ramezanpour, K., Ampadu, P., Diehl, W.: Rs-mask: Random space masking as an integrated countermeasure against power and fault analysis. In: 2020 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2020, San Jose, CA, USA, December 7-11, 2020. pp. 176–187. IEEE (2020). https://doi.

org/10.1109/HOST45689.2020.9300266, https://doi.org/10.1109/HOST45689.2020.9300266

42. Rasoolzadeh, S., Shahmirzadi, A.R., Moradi, A.: Impeccable circuits III. In: IEEE International Test Conference, ITC 2021, Anaheim, CA, USA, October 10-15, 2021. pp. 163–169. IEEE (2021). https://doi.org/10.1109/ITC50571.2021.00024, https://doi.org/10.1109/ITC50571.2021.00024
43. Reparaz, O., Meyer, L.D., Bilgin, B., Arribas, V., Nikova, S., Nikov, V., Smart, N.P.: CAPA: the spirit of beaver against physical attacks. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10991, pp. 121–151. Springer (2018). https://doi.org/10.1007/978-3-319-96884-1_5, https://doi.org/10.1007/978-3-319-96884-1_5
44. Richter-Brockmann, J., Güneysu, T.: Improved side-channel resistance by dynamic fault-injection countermeasures. In: 31st IEEE International Conference on Application-specific Systems, Architectures and Processors , ASAP 2020, Manchester, United Kingdom, July 6-8, 2020. pp. 117–124. IEEE (2020). https://doi.org/10.1109/ASAP49362.2020.00029, https://doi.org/10.1109/ASAP49362.2020.00029
45. Richter-Brockmann, J., Sasdrich, P., Güneysu, T.: Revisiting fault adversary models - hardware faults in theory and practice. IEEE Trans. Computers **72**(2), 572–585 (2023). https://doi.org/10.1109/TC.2022.3164259, https://doi.org/10.1109/TC.2022.3164259
46. Roche, T., Lomné, V., Khalfallah, K.: Combined fault and side-channel attack on protected implementations of AES. In: Prouff, E. (ed.) Smart Card Research and Advanced Applications - 10th IFIP WG 8.8/11.2 International Conference, CARDIS 2011, Leuven, Belgium, September 14-16, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7079, pp. 65–83. Springer (2011). https://doi.org/10.1007/978-3-642-27257-8_5, https://doi.org/10.1007/978-3-642-27257-8_5
47. Saha, S., Bag, A., Jap, D., Mukhopadhyay, D., Bhasin, S.: Divided we stand, united we fall: Security analysis of some SCA+SIFA countermeasures against sca-enhanced fault template attacks. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13091, pp. 62–94. Springer (2021). https://doi.org/10.1007/978-3-030-92075-3_3, https://doi.org/10.1007/978-3-030-92075-3_3
48. Saha, S., Bag, A., Roy, D.B., Patranabis, S., Mukhopadhyay, D.: Fault template attacks on block ciphers exploiting fault propagation. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12105, pp. 612–643. Springer (2020). https://doi.org/10.1007/978-3-030-45721-1_22, https://doi.org/10.1007/978-3-030-45721-1_22
49. Saha, S., Jap, D., Roy, D.B., Chakraborti, A., Bhasin, S., Mukhopadhyay, D.: Transform-and-encode: A countermeasure framework for statistical ineffective fault attacks on block ciphers. IACR Cryptol. ePrint Arch. p. 545 (2019), https://eprint.iacr.org/2019/545

50. Schellenberg, F., Finkeldey, M., Gerhardt, N., Hofmann, M., Moradi, A., Paar, C.: Large laser spots and fault sensitivity analysis. In: Robinson, W.H., Bhunia, S., Kastner, R. (eds.) 2016 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2016, McLean, VA, USA, May 3-5, 2016. pp. 203–208. IEEE Computer Society (2016). https://doi.org/10.1109/HST.2016.7495583, https://doi.org/10.1109/HST.2016.7495583
51. Schneider, T., Moradi, A., Güneysu, T.: Parti - towards combined hardware countermeasures against side-channel and fault-injection attacks. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 302–332. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_11, https://doi.org/10.1007/978-3-662-53008-5_11
52. Shahmirzadi, A.R., Rasoolzadeh, S., Moradi, A.: Impeccable circuits II. In: 57th ACM/IEEE Design Automation Conference, DAC 2020, San Francisco, CA, USA, July 20-24, 2020. pp. 1–6. IEEE (2020). https://doi.org/10.1109/DAC18072.2020.9218615, https://doi.org/10.1109/DAC18072.2020.9218615
53. Simon, T., Batina, L., Daemen, J., Grosso, V., Massolino, P.M.C., Papagiannopoulos, K., Regazzoni, F., Samwel, N.: Friet: An authenticated encryption scheme with built-in fault detection. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12105, pp. 581–611. Springer (2020). https://doi.org/10.1007/978-3-030-45721-1_21, https://doi.org/10.1007/978-3-030-45721-1_21
54. Skorobogatov, S.P., Anderson, R.J.: Optical fault induction attacks. In: Jr., B.S.K., Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2523, pp. 2–12. Springer (2002). https://doi.org/10.1007/3-540-36400-5_2, https://doi.org/10.1007/3-540-36400-5_2
55. Spruyt, A., Milburn, A., Chmielewski, L.: Fault injection as an oscilloscope: Fault correlation analysis. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(1), 192–216 (2021). https://doi.org/10.46586/tches.v2021.i1.192-216, https://doi.org/10.46586/tches.v2021.i1.192-216
56. Vafaei, N., Zarei, S., Bagheri, N., Eichlseder, M., Primas, R., Soleimany, H.: Statistical effective fault attacks: The other side of the coin. IEEE Trans. Inf. Forensics Secur. **17**, 1855–1867 (2022). https://doi.org/10.1109/TIFS.2022.3172634, https://doi.org/10.1109/TIFS.2022.3172634
57. Yen, S., Joye, M.: Checking before output may not be enough against fault-based cryptanalysis. IEEE Trans. Computers **49**(9), 967–970 (2000). https://doi.org/10.1109/12.869328, https://doi.org/10.1109/12.869328
58. Zussa, L., Dutertre, J., Clédière, J., Tria, A.: Power supply glitch induced faults on FPGA: an in-depth analysis of the injection mechanism. In: 2013 IEEE 19th International On-Line Testing Symposium (IOLTS), Chania, Crete, Greece, July 8-10, 2013. pp. 110–115. IEEE (2013). https://doi.org/10.1109/IOLTS.2013.6604060, https://doi.org/10.1109/IOLTS.2013.6604060