How Practical are Fault Injection Attacks, Really?

Jakub Breier[1] and Xiaolu Hou[2]
[1]Silicon Austria Labs, Graz, Austria
[2]Slovak University of Technology, Bratislava, Slovakia
jbreier@jbreier.com; houxiaolu.email@gmail.com

Abstract

Fault injection attacks (FIA) are a class of active physical attacks, mostly used for malicious purposes such as extraction of cryptographic keys, privilege escalation, attacks on neural network implementations. There are many techniques that can be used to cause the faults in integrated circuits, many of them coming from the area of failure analysis. In this paper we tackle the topic of practicality of FIA. We analyze the most commonly used techniques that can be found in the literature, such as voltage/clock glitching, electromagnetic pulses, lasers, and Rowhammer attacks. To summarize, FIA can be mounted on most commonly used architectures from ARM, Intel, AMD, by utilizing injection devices that are often below the thousand dollar mark. Therefore, we believe these attacks can be considered practical in many scenarios, especially when the attacker can physically access the target device.

1 Introduction

Cryptographic algorithms, both symmetric and public key, are susceptible to fault injection attacks (FIA). In 1997, Boneh, DeMillo and Lipton showed that an implementation of RSA using Chinese remainder theorem (CRT) can be easily broken by using faults [17]. In the same year, Biham and Shamir published an attack titled *differential fault analysis* (DFA) that can break most of the symmetric cryptosystems [15]. The working principle of FIA is simple – the attacker injects a fault during the algorithm execution, and then, based on the analysis method, they utilize the information from the faulted execution to narrow down the search space of the secret/private key. Nowadays, 25 years after these attacks were published, this area has become one of the major areas in hardware security, alongside the passive side-channel attacks (SCA) [64]. Many analysis methods have been published to date, to mention the most prominent ones apart from the DFA: statistical ineffective fault analysis (SIFA) [37], persistent fault attack (PFA) [96], fault sensitivity analysis (FSA) [60], fault template attacks (FTA) [80], and FIA combined with SCA [72]. Aside from targeting cryptography, fault attacks have been used for bypassing checking routines [94, 35], and even faulting neural network implementations [24, 28]. Various methods have been used for injecting faults, from clock/voltage glitches [19], to electromagnetic pulses [65], to lasers [26], to X-rays [5], to Rowhammer attacks [67].

While there have been several surveys [41, 54, 7] and book publications [49, 72, 23] summarizing the state-of-the-art in the area of FIA, there is an important question that often remains unanswered. It is natural that whenever someone from the outside of this area comes across a work that details an attack on some implementations, they wonder whether such an attack vector can be realized in a real world, not just an expensive laboratory setting with a highly skilled personnel. In this paper, we try to address this issue and provide an answer to:

"How practical are fault injection attacks?"

We tackle this question from multiple points of view – cost of equipment, remote access, device decapsulation, precision of the fault, and device architecture. We note that this article is not a comprehensive survey

Table 1: Overview of the techniques currently available in the literature with the lowest cost for a given target device and a fault model. A *"low"* cost means that only a standard desktop PC (and in some cases, connection wires) are needed for the attack.

Target device	Fault model	Remote	Method with lowest cost		
			Reference	Technique	Cost
AVR	bit flip	no	[3]	optical (laser)	∼100K USD
	bit set/reset	no	[12]	EM	30K USD
	random byte	no	[43]	optical (flashgun)	500 EUR
	instruction skip	no	[36]	EM	10 USD
ARM (standalone)	bit set/reset	no	[65]	EM	30K USD
	random byte	no	[43]	optical (flashgun)	500 EUR
	instruction skip	no	[34]	clock glitch	130 USD
ARM (embedded)	bit flip	no	[94]	optical (laser)	∼100K USD
	random byte	yes	[90]	voltage glitch	low
	instruction skip	no	[36]	EM	10 USD
FPGA	bit flip	no	[47]	optical (laser)	100K USD
	bit set/reset	no	[71]	EM	∼30K USD
	random byte	no	[53]	voltage glitch	∼300 USD
	execution faults	yes	[4]	temperature/voltage	low
Intel	random byte	no	[33]	voltage glitch	30 USD
		yes	[77]	voltage glitch	low
AMD	random byte	no	[30]	voltage glitch	30 USD
DRAM	bit flip	yes	[55]	Rowhammer	low
TRNG	stuck-at fault	no	[62]	EM	∼30K USD

of all the works in the area – we select works that provide a reasonable description of the experimental setup that can be used for a proper comparison.

The rest of this paper is organized as follows. Section 2 provides an overview of the cost of each achievable fault model published so far. Section 3 gives a detailed information on each commonly used fault injection technique. Section 4 provides a discussion on countermeasures and future work, and finally, Section 5 concludes this work.

2 Current State-of-the-Art Techniques and Their Practicality

From the attacker's point of view, a natural question is *"I have a target device and a desired fault model, what are the possible ways of achieving the fault and what is the cost?"* In this section, we aim at answering this by listing the available works along with the details that are important for the attacker.

Generally, the following categories of fault models are used in the analysis methods in the literature:

- **Bit flip** is the change of the bit value to the opposite value, while this bit can be precisely selected by the attacker. A multiple bit flips also fall within in this category as long as all the target bits are selected by the attacker. For example, most of the fault attacks on neural networks utilize this model [78, 28].
- **Bit set/reset** is the change of the bit value either to '1' (set) or to '0' (reset). Again, the assumption is that the attacker can select the bit to be set/reset. This fault model is very powerful and can be utilized for example for blind fault attacks [57].
- **Random byte** is a less precise fault model where a value of a particular byte changes to some random value. This is considered to be the most relaxed fault model to achieve a successful DFA attack [40,

61].

- **Instruction skip** practically ignores the execution of the currently processed instruction. Powerful attacks can be introduced by using this fault model, such as privilege escalation [92], a simple key extraction [27], or a neural network misclassification [48].
- **Execution faults** occur in FPGAs where the values being processed are affected by setup violations. For example, physically unclonable functions can be attacked with this fault model [89].
- **Stuck-at faults** permanently changes the value of the stored data into some other value. SIFA can be used with this fault model [37], and also, true random number generators (TRNGs) can be biased by using stuck-at faults [62].

The high-level overview of current techniques is listed in Table 1[1]. We aimed at finding the techniques with the lowest cost for the given target device and the fault model, along with the information whether this attack can be carried out remotely. We believe that when designing a fault analysis method, it is important to know whether it can be carried out in practice and therefore, the table provides a sufficient answer to that. There are several additional remarks that we would like to mention:

- Wherever we use the tilde character ('∼'), we estimate the cost based on the information on the used setup. Generally a working setup for an electromagnetic fault injection (EMFI) can be assembled for around 30K USD, and for a laser fault injection (LFI) for around 100K USD. If there is no tilde, the number was taken directly from the referenced paper.
- In case of ARM, we distinguish between a standalone chip and an embedded one. Generally, the non-remote techniques should be usable for both cases, however, the remote attack assumes a complex operating system (e.g. Linux).
- In the first four categories, it is important to know in which component the attack happened. For example, an attack in the register would only have a very short time effect, the change in the SRAM would generally have a longer effect (and can be used for example for a persistent fault analysis [96]), while the fault in the flash would affect the program itself. Below we provide the details for the affected device categories:
  - AVR: [3] and [43] target the SRAM while [12] aims at the flash memory.
  - ARM (standalone): [65] targets the flash memory and [43] corrupts the registers.
  - ARM (embedded): [94] targets the registers.
  - FPGA: [47] and [71] attack the registers, and [53] causes the setup violations corrupting the processed data. The execution faults presented by [4] are also caused by setup violations.
- A remote voltage glitch attack, Plundervolt [66], also achieved a certain bit flip fault models. However, the bits flipped could not be chosen by the attacker, only certain bits at specific locations could be flipped.
- When the cost is indicated as "low", we mean that only a standard desktop PC (and in some cases, connection wires) are needed for the attack.

In the remaining part of this paper we present each technique in more detail as the aim of this section was to provide a general overview.

3 Detailed Overview of Fault Injection Techniques

In this section, we will detail the most popular fault injection techniques that are used for testing cryptographic devices nowadays.

[1] The table was populated by crawling through the available works. If you have published a work that should be listed, please contact us and we will update the live version of the paper accessible at `https://eprint.iacr.org/2022/301`

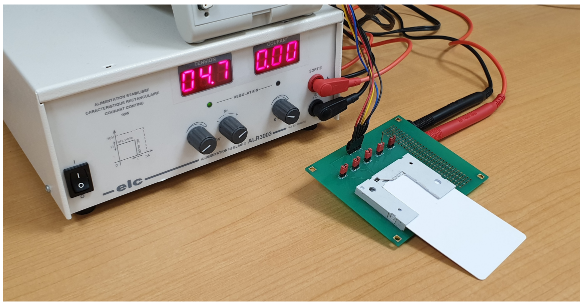


Figure 1: An example of a voltage glitch on a smart card.

3.1 Clock/Voltage Glitching

Voltage and clock manipulation based fault injection methods are low-cost, and generally, no sophisticated equipment is necessary. They can be achieved both remotely and with target device in hand.

With physical access to the device, voltage glitching is done by manipulating the power supply, causing the faulty behavior on a device. It can be achieved by creating precise high variations in a power supply or by under-powering the device.

Precise high variations, or power spikes, modify the state of latches of flip-flops, influencing the control and data path logic of the circuit [58]. For example, if the voltage spike happens during memory reading, wrong data may be retrieved. It was also shown that different shape of the glitch waveform affects the success of the attack [19]. Under-powering of the device can cause erroneous output. Such method affects the algorithm continuously and might cause faults throughout the computation. But single faults are possible when the insufficient power supply causes gentle enough stress so that dysfunctions do not occur immediately after the computation starts and multi-faults do not happen [85]. Figure 1 depicts a real voltage glitch attack based on under-powering on smart cards.

When the attacker has access to the target device, voltage glitching is generally easy to implement and it is the cheapest fault injection method as the necessary equipment are wires for connecting to the device and a power source. On the other hand, this method requires that the attacker has access to the power supply line of the device.

Voltage glitching attacks were even used to break security enclaves of Intel [33] and AMD [30]. Both attacks used an inexpensive Teensy 4.0 board [2] (≈ 30 USD), making them highly practical in terms of equipment cost. Naturally, for such attacks it is necessary to have a deep knowledge of the attacked architecture.

Another inexpensive fault injection method is a clock glitch. Computation devices use external or internal clocks to synchronize all of their calculations. When the clock signal is changed, the resulting computation might have wrong instruction executed or data corrupted. For devices that require an external clock generator, the fault can be introduced by supplying a bad clock signal, e.g. a signal that contains fewer pulses than the normal one [50]. Devices with internal clock generators, however, cannot be attacked by a clock glitching method.

Clock glitches are generally considered as the simplest fault injection method as the attack devices are easy to operate with. For example, clock glitches can be achieved by using low-end field-programmable gate array (FPGA) boards [10, 38]. Recently, a multifault evaluation platform named TRAITOR with a price below 130 USD was proposed in [34].

For clock glitches, the adversary needs to have a direct control over the clock generator, which is a common scenario when attacking smart cards.

When it comes to remote attacks, clock/voltage glitching can also be achieved. A relatively new class of fault attacks reveals vulnerabilities following the advancement of efficient energy management. The

[2] https://www.pjrc.com/store/teensy40.html

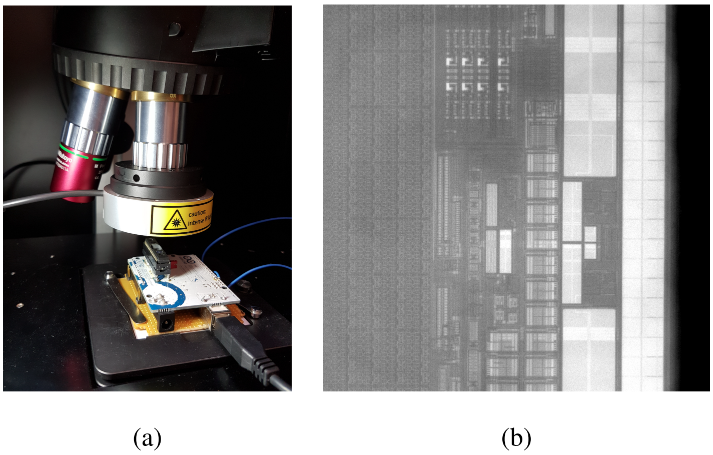

(a) (b)

Figure 2: Optical fault injection attacks: (a) pulsed laser fault injection on ATmega328P mounted on a modified Arduino UNO board as a target; (b) usage of the same setup to get an infrared image of the chip.

designers of energy management rarely consider the security aspect due to the complexity of devices from hardware point of view as well as software executed, cost and time-to-market constraint [75]. By exploiting Dynamic Voltage & Frequency Scaling (DVFS), Tang et al. [90] developed CLKSCREW, where the attacker can manipulate the frequency and voltage of an Nexus 6 phone, forcing the processor to operate beyond recommended limits. They experimentally verified that one-byte random fault is achievable. CLKSCREW can be achieved only by software control of energy management hardware regulators in the target devices. Similar vulnerabilities were also exploited in ARM-based Krait processor from a commodity Android [76] and intel SGX [77].

The features of those attacks are that they are software-based attacks, hence allowing the threat model to shift from a local attacker to a potentially remote attacker. More and more software-based fault attacks by voltage glitching were later developed, e.g. [52, 66]

3.2 Optical Fault Injection

The phenomenon of ionization effects on transistors has been known for decades. The usage of lasers in the area of reliability of microchips is a standard way to test their robustness and dates back to the very beginning of the computing era [44]. It is especially important to test chips that will be deployed in adverse conditions. For example, it was shown that the flip-flop circuits in the satellites are affected by cosmic rays [16]. It was just a matter of time until the first optical fault injection technique is used in the area of cryptography after it was discovered that faults can compromise the security [87].

Optical fault injection area is perhaps the most diverse from the listed techniques. On one hand, there are works using an inexpensive camera flash to cause random faults [87], on the other, an attacker can use a nanofocused X-ray beam to target a single transistor [5]. Moreover, it was shown that with the usage of lasers it is possible to probe the memory without changing it, which can reveal its content [32]. Therefore, the practicality range varies greatly for this class of attacks.

When it comes to security evaluation labs, the method of choice would be a laser fault injection (LFI). There are numerous companies selling out-of-the-box setups for performing LFI. A standard setup would consist of the following parts: laser source, objective lens, motorized positioning table, and a controlling device. A digital oscilloscope can be used to precisely align the laser activation with the execution of the target routine on the device. Normally, there would be an optical splitter so that an infrared (IR) camera could be included on the same lens. Such a setup is depicted in Figure 2(a), with a backside chip surface picture taken from the IR camera in Figure 2(b).

While the cost of a fully assembled setup would be normally south of 50k USD, recently there has been a proposal showing that it is possible to assemble a working setup under 500 USD [51]. The authors used a solid state laser diode allowing a pulse repetition rate of 200 MHz which is on par with expensive setups from established testing equipment companies.

However, as mentioned earlier, lasers are not the only method within the optical fault injection area. The very first paper in the security realm showed that by using a camera flash coupled with a 1500× magnifying lens (mounted on Wentworth Labs MP-901 manual prober), it was possible to change the value of a single SRAM on a PIC16 chip. While one could argue that the price of such a manual prober could be relatively high, a more recent paper has shown that it is possible to use an inexpensive ball lens to focus the camera flash [43]. Such a setup was used to target registers and skip instructions on ARM Cortex-M0, and to change the values in the RAM and skip instructions on ATmega328P.

Optical fault injection is considered as a semi-invasive attack technique, meaning that the chip package needs to be removed to expose the chip to the optical source. This is the main drawback as sometimes it is not possible to de-package the chip without damaging the circuitry or the bonding wires. The injection is normally done on the backside of the chip, as the components are protected from the front side. This creates another challenge as the absorption depth of silicon varies for different wavelenghts, and therefore, the silicon substrate might need to be thinned down to allow an attack. Either a mechanical or a chemical decapsulation techniques can be used to remove the package, each offering different set of advantages and disadvantages [21]. For thinning the substrate, a mechanical delayering is necessary, often involving expensive devices (e.g. UltraTec ASAP-I was used in [22]). However, if the chip can be properly prepared, optical fault injection offers a very precise and repeatable way to induce errors [3].

There are several other fault injection techniques which are somewhat related to optical techniques in their modus operandi. There is a long history of using electron and ion beam techniques in the area of failure analysis for reliability testing of integrated circuits [88]. To the best of our knowledge, the usage of X-ray nanobeams was the only work within this realm used for security analysis [5]. The advantage of this method is that there is no need to remove the chip package as it is transparent to the beams. These techniques range in millions of USD and are out of the practical bounds for the class of attackers normally considered when attacking devices such as credit cards, IoT devices, etc. However, a consideration needs to be in place for very critical systems such as military communication equipment.

To summarize, optical fault injection techniques offer a high precision and repeatability at a relatively high cost, apart from few exceptions. The chip preparation is the main drawback of these techniques (unless the very expensive methods are used), and often makes it impractical to use outside of laboratory environment. As it is often useful to assume highly motivated attackers with high capabilities, laser fault injection is a de-facto standard for security testing labs that certify security critical elements.

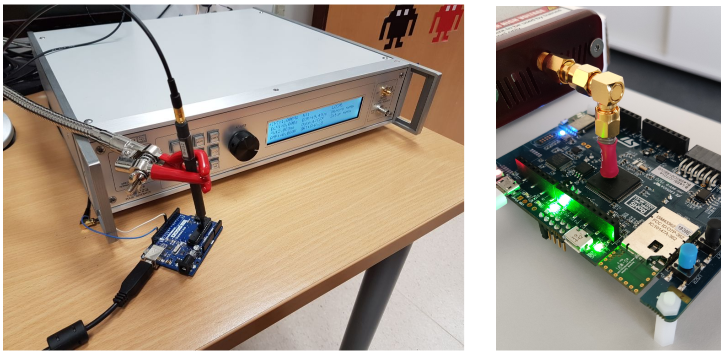
3.3 Electromagnetic Fault Injection

Cryptographic circuits are usually a combination of digital logic, implementing the algorithm, and analog logic which handles the clock sybsystem and random number generators. Electromagnetic (EM) emanation affects both analog and digital blocks, despite their different physical characteristics. However, a different approach needs to be taken in each case.

Analog blocks are vulnerable to powerful harmonic EM waves. The attacker generates a stable sinusoidal signal at a given frequency that injects a harmonic wave creating a parasitic signal [45]. Such a signal can bias the clock behavior or inject an additional power directly and locally into the chip. Equipment for this type of EM injection usually consists of a motorized positioning table, signal generation module, and an oscilloscope.

Digital blocks are clocked, therefore the preferable way to disrupt their behavior is via EM pulse injection capable of injecting faults in a specific clock cycle in a controllable way [82]. The aim is to inject a sudden and sharp EM pulse into the integrated circuit, introducing intense transient currents altering the behavior of logic cells. Generally, the equipment consists of a high voltage pulse generator and a coil with a ferrite core, serving as an injection probe. An example of such an equipment is depicted in Figure 3.

As the fault analysis methods mostly work with data perturbation (bit flips, bit sets/resets, random faults,



(a) (b)

Figure 3: Pulse EM injection in practice: (a) a high voltage EM pulse generator inducing faults through an off-the-shelf injection probe into ATmega328P (Arduino UNO board); (b) a compact EM pulse generator injecting faults through a custom made injection probe into ARM Cortex-M4 (STM32 Discovery board).

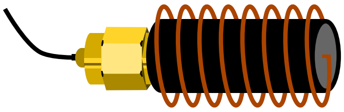


Figure 4: A generic depiction of an EM fault injection probe.

etc.), pulse EM injection is more prevalent in the literature. This injection method provides a good trade-off between the cost and the precision. Pulse injectors can be bought for a relatively inexpensive price, for example, NewAE sells their ChipSHOUTER for $\approx$ 3.3k USD[3] (used for example in [68] to break hardware wallets). For more powerful and precise equipment, one can look into Avtech pulse generators that would generally range between 10k - 20k USD[4]. A near-field injection probe can either be bought (cost here would be a couple of hundred USD) or manufactured from very low-cost components. Several research articles explore the possibility of a custom probe design [70, 81, 13]. Generally, a ferrite core, a copper wire, a connector, and a heat shrinking tube are enough to create a custom probe (depicted in Figure 4).

Recently, there have been published several custom-made low-cost EMFI device prototypes which can be easily reproduced by using inexpensive off-the-shelf components and a moderate knowledge in electronics. BADFET [35] was shown to be capable of overcoming a secure boot, SiliconToaster [1] was used to defeat a firmware security protection of an IoT device, and another low-cost device was shown to be effective in privilege escalation [36].

EM fault injection does not need a device decapsulation for chips enclosed in a standard epoxy package, which is one of the main drawbacks of laser fault injection. The advantage over the clock/voltage glitching is that there is no need to attach any wires on the power supply.

To summarize, EM fault injection is a highly practical technique for attackers that have a possession of the target device – it offers good fault reproducibility and precision at a relatively low cost.

3.4 Rowhammer Attacks

The earliest remote fault injection was based on Rowhammer attack [55], which exploits the physical characteristics of DRAM – by aggressively reading/writing to some address in DRAM, the attacker can flip bits in a nearby memory location. Such a vulnerability is mostly due to the advancing of DRAM manufacturing

[3] https://www.newae.com/chipshouter

[4] https://www.avtechpulse.com/medium/

technology, which allowed smaller cells to be placed closer to each other. A smaller cell also means less capacity for charge, hence lower noise margin and making the cell more vulnerable to data loss [63]. High density of cells additionally causes electromagnetic coupling effects between them, resulting in unwanted interactions [56].

Rowhammer attack has been demonstrated on various platforms: browsers [18, 42, 84], cloud environment [69, 79, 95], smartphones [93, 39] and flash storage [31, 59]. These attacks do not require the attacker to have a physical access to the device except for the ability to execute code on the target device. Tatar et al. [91] demonstrated that Rowhammer can also be carried out by sending network packets to a target machine connected to RDMA-enabled networks.

In terms of the equipment, to achieve Rowhammer attacks, the attacker just needs an access to Internet and a computer. A deeper knowledge of computer architecture might be required for more sophisticated attacks.

4 Discussion

4.1 Countermeasures

While the focus of this paper is not on countermeasures, the existence of those confirms that fault attacks constitute a threat against security-critical implementations. The following techniques have been proposed up to date:

- **Redundancy.** Various usage of redundancy can be implemented to protect against different fault models. The most basic technique would be a duplication where the same circuit is deployed twice and there is an integrity check. In terms of software implementations, this can be achieved by running the same execution twice in series (or in parallel on multiple processors). A triplication with a majority voting can be used against more sophisticated attacks such as SIFA [29]. Intra-instruction redundancy was shown to be capable of protecting against instruction skips [74]. Construction of various codes can be utilized for multiple bit corruptions within the same data [25]. Redundant hardware circuits were proposed to detect faults [83, 2].
- **Sensors.** Device-level sensors can be used to detect fault injections [11, 6]. Glitch detectors have been used to raise an alert when there is a sudden change in the EM field [97, 20]. Similar sensors have been shown to be efficient against laser fault injection [46]. In that direction, it is also possible to use various sensors to detect de-packaging of the chip – for example, a light sensor, or a simple wire mesh in the epoxy resin that becomes non conductive when the package is tampered with.
- **Algorithmic techniques.** Another direction to thwart FIA is to propose an algorithm design that offers inherent fault detection. This is a relatively new area, started with a lightweight block cipher CRAFT [14], and followed by an authenticated block cipher FRIET [86]. While the two above-mentioned ciphers relied on usage of coding theory, the most recent approach, a lightweight block cipher DEFAULT [9], utilized linear structures introduced in otherwise non-linear substitution components of the algorithm. Generally, this type of countermeasure seems to be getting traction as it offers a clear advantage of unburdening the implementer from dealing with the fault protection.

There are also other types of countermeasures that do not fall within these categories, such as infective techniques [73] or protocol-level countermeasures [8].

All of the countermeasures naturally introduce an overhead, either in power consumption, time, or space. It is therefore necessary to conduct some sort of a risk assessment to be able to choose the right level of protection depending on the value of assets and potential threat vectors.

4.2 Future Directions

There are several trends emerging in the recent literature that can be identified as the next directions in the area of fault injection techniques:

- **Techniques to break security enclaves.** Very recent voltage attacks have been shown effective against security enclaves of both main PC processor manufacturers, Intel [33] and AMD [30]. ARM Trustzone was even broken by a remote attack manipulating the operating frequency [90, 76]. We believe this area will gain a serious traction in the next few years as the security implications of attacking PCs and smartphones are a concern for general public.
- **Low-cost fault injection techniques.** As the fault injection is moving from academic environment and evaluation labs to hardware security enthusiasts and hackers, there is a push towards affordable fault injection techniques. EMFI [36], optical [43], and also voltage glitch [34] custom-made equipment can be built with standard components ranging in a few hundreds of dollars. It is expected that researchers will continue building inexpensive devices while tweaking their precision and ease-of-use.
- **Remote attacks.** As shown in Table 1, remote attacks are missing for the majority of fault models and target devices. Recent works, however, are starting to fill this gap. The most popular direction is the development of software-based fault attacks [52, 66, 77, 90]. Due to the attack method nature, among the techniques we describe here, only voltage/clock glitches and Rowhammer are achievable remotely. Making it possible to remotely target some device with a fault attack creates a very potent threat as these attacks are rarely considered in the security risk assessment. Therefore, there is a strong motivation for researchers to find novel ways to disturb devices by faults remotely.

5 Conclusion

In this paper we aimed at analyzing the practicality of fault injection attacks in a real world setting. For a target device and a desired fault model, we listed the method with the lowest cost from the literature. Additionally, we provided a short survey on different fault injection techniques, listing the current state-of-the-art for each area. The results demonstrate that a reasonable amount of faults can be achieved with affordable cost for individual attackers and hence can be considered very practical.

References

[1] K. M. Abdellatif and O. Hériveaux. Silicontoaster: a cheap and programmable em injector for extracting secrets. In *2020 Workshop on Fault Detection and Tolerance in Cryptography (FDTC)*, pages 35–40. IEEE, 2020.

[2] A. Aghaie, A. Moradi, S. Rasoolzadeh, A. R. Shahmirzadi, F. Schellenberg, and T. Schneider. Impeccable circuits. *IEEE Transactions on Computers*, 69(3):361–376, 2019.

[3] M. Agoyan, J.-M. Dutertre, A.-P. Mirbaha, D. Naccache, A.-L. Ribotta, and A. Tria. How to flip a bit? In *2010 IEEE 16th International On-Line Testing Symposium*, pages 235–239. IEEE, 2010.

[4] M. M. Alam, S. Tajik, F. Ganji, M. Tehranipoor, and D. Forte. Ram-jam: Remote temperature and voltage fault attack on fpgas using memory collisions. In *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 48–55. IEEE, 2019.

[5] S. Anceau, P. Bleuet, J. Clédière, L. Maingault, J.-l. Rainard, and R. Tucoulou. Nanofocused x-ray beam to reprogram secure circuits. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 175–188. Springer, 2017.

[6] M. T. H. Anik, J.-L. Danger, S. Guilley, and N. Karimi. Detecting failures and attacks via digital sensors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(7):1315–1326, 2020.

[7] A. Baksi, S. Bhasin, J. Breier, D. Jap, and D. Saha. Fault attacks in symmetric key cryptosystems. *Cryptology ePrint Archive*, 2020.

[8] A. Baksi, S. Bhasin, J. Breier, M. Khairallah, and T. Peyrin. Protecting block ciphers against differential fault attacks without re-keying. In *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 191–194. IEEE, 2018.

[9] A. Baksi, S. Bhasin, J. Breier, M. Khairallah, T. Peyrin, S. Sarkar, and S. M. Sim. Default: Cipher level resistance against differential fault attack. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 124–156. Springer, 2021.

[10] J. Balasch, B. Gierlichs, and I. Verbauwhede. An in-depth and black-box characterization of the effects of clock glitches on 8-bit mcus. In *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 105–114. IEEE, 2011.

[11] R. P. Bastos, F. S. Torres, J.-M. Dutertre, M.-L. Flottes, G. Di Natale, and B. Rouzeyre. A bulk built-in sensor for detection of fault attacks. In *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 51–54. IEEE, 2013.

[12] A. Beckers, J. Balasch, B. Gierlichs, I. Verbauwhede, S. Osuka, M. Kinugawa, D. Fujimoto, and Y. Hayashi. Characterization of em faults on atmega328p. In *2019 Joint International Symposium on Electromagnetic Compatibility, Sapporo and Asia-Pacific International Symposium on Electromagnetic Compatibility (EMC Sapporo/APEMC)*, pages 1–4. IEEE, 2019.

[13] A. Beckers, M. Kinugawa, Y. Hayashi, D. Fujimoto, J. Balasch, B. Gierlichs, and I. Verbauwhede. Design considerations for em pulse fault injection. In *International Conference on Smart Card Research and Advanced Applications*, pages 176–192. Springer, 2019.

[14] C. Beierle, G. Leander, A. Moradi, and S. Rasoolzadeh. Craft: lightweight tweakable block cipher with efficient protection against dfa attacks. *IACR Transactions on Symmetric Cryptology*, 2019(1):5–45, 2019.

[15] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In *Annual international cryptology conference*, pages 513–525. Springer, 1997.

[16] D. Binder, E. C. Smith, and A. Holman. Satellite anomalies from galactic cosmic rays. *IEEE Transactions on Nuclear Science*, 22(6):2675–2680, 1975.

[17] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults. In *International conference on the theory and applications of cryptographic techniques*, pages 37–51. Springer, 1997.

[18] E. Bosman, K. Razavi, H. Bos, and C. Giuffrida. Dedup est machina: Memory deduplication as an advanced exploitation vector. In *2016 IEEE symposium on security and privacy (SP)*, pages 987–1004. IEEE, 2016.

[19] C. Bozzato, R. Focardi, and F. Palmarini. Shaping the glitch: optimizing voltage fault injection attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 199–224, 2019.

[20] J. Breier, S. Bhasin, and W. He. An electromagnetic fault injection sensor using hogge phase-detector. In *2017 18th International Symposium on Quality Electronic Design (ISQED)*, pages 307–312. IEEE, 2017.

[21] J. Breier and C.-N. Chen. On determining optimal parameters for testing devices against laser fault attacks. In *2016 International Symposium on Integrated Circuits (ISIC)*, pages 1–4. IEEE, 2016.

[22] J. Breier, W. He, S. Bhasin, D. Jap, S. Chef, H. G. Ong, and C. L. Gan. Extensive laser fault injection profiling of 65 nm fpga. *Journal of Hardware and Systems Security*, 1(3):237–251, 2017.

[23] J. Breier, X. Hou, and S. Bhasin. *Automated Methods in Cryptographic Fault Analysis*. Springer, 2019.

[24] J. Breier, X. Hou, D. Jap, L. Ma, S. Bhasin, and Y. Liu. Practical fault attack on deep neural networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2204–2206, 2018.

[25] J. Breier, X. Hou, and Y. Liu. On evaluating fault resilient encoding schemes in software. *IEEE Transactions on Dependable and Secure Computing*, 18(3):1065–1079, 2019.

[26] J. Breier and D. Jap. Testing feasibility of back-side laser fault injection on a microcontroller. In *Proceedings of the WESS'15: Workshop on Embedded Systems Security*, pages 1–6, 2015.

[27] J. Breier, D. Jap, and C.-N. Chen. Laser profiling for the back-side fault attacks: with a practical laser skip instruction attack on aes. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, pages 99–103, 2015.

[28] J. Breier, D. Jap, X. Hou, S. Bhasin, and Y. Liu. Sniff: reverse engineering of neural networks with fault attacks. *IEEE Transactions on Reliability*, 2021.

[29] J. Breier, M. Khairallah, X. Hou, and Y. Liu. A countermeasure against statistical ineffective fault analysis. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(12):3322–3326, 2020.

[30] R. Buhren, H.-N. Jacob, T. Krachenfels, and J.-P. Seifert. One glitch to rule them all: Fault injection attacks against amd's secure encrypted virtualization. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2875–2889, 2021.

[31] Y. Cai, S. Ghose, Y. Luo, K. Mai, O. Mutlu, and E. F. Haratsch. Vulnerabilities in mlc nand flash memory programming: Experimental analysis, exploits, and mitigation techniques. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 49–60. IEEE, 2017.

[32] S. Chef, C. Chua, J. Tay, Y. Siah, S. Bhasin, J. Breier, and C. Gan. Descrambling of embedded sram using a laser probe. In *2018 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pages 1–6. IEEE, 2018.

[33] Z. Chen, G. Vasilakis, K. Murdock, E. Dean, D. Oswald, and F. D. Garcia. {VoltPillager}: Hardware-based fault injection attacks against intel {SGX} enclaves using the {SVID} voltage scaling interface. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 699–716, 2021.

[34] L. Claudepierre, P.-Y. Péneau, D. Hardy, and E. Rohou. Traitor: a low-cost evaluation platform for multifault injection. In *Proceedings of the 2021 International Symposium on Advanced Security on Software and Systems*, pages 51–56, 2021.

[35] A. Cui and R. Housley. {BADFET}: Defeating modern secure boot using {Second-Order} pulsed electromagnetic fault injection. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.

[36] S. Delarea and Y. Oren. Practical, low-cost fault injection attacks on personal smart devices. *Applied Sciences*, 12(1):417, 2022.

[37] C. Dobraunig, M. Eichlseder, T. Korak, S. Mangard, F. Mendel, and R. Primas. Sifa: exploiting ineffective fault inductions on symmetric cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 547–572, 2018.

[38] S. Endo, T. Sugawara, N. Homma, T. Aoki, and A. Satoh. An on-chip glitchy-clock generator for testing fault injection attacks. *Journal of Cryptographic Engineering*, 1(4):265–270, 2011.

[39] P. Frigo, C. Giuffrida, H. Bos, and K. Razavi. Grand pwning unit: Accelerating microarchitectural attacks with the gpu. In *2018 ieee symposium on security and privacy (sp)*, pages 195–210. IEEE, 2018.

[40] C. Giraud. Dfa on aes. In *International Conference on Advanced Encryption Standard*, pages 27–41. Springer, 2004.

[41] C. Giraud and H. Thiebeauld. A survey on fault attacks. In *Smart Card Research and Advanced Applications VI*, pages 159–176. Springer, 2004.

[42] D. Gruss, C. Maurice, and S. Mangard. Rowhammer. js: A remote software-induced fault attack in javascript. In *International conference on detection of intrusions and malware, and vulnerability assessment*, pages 300–321. Springer, 2016.

[43] O. M. Guillen, M. Gruber, and F. D. Santis. Low-cost setup for localized semi-invasive optical fault injection attacks. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 207–222. Springer, 2017.

[44] D. H. Habing. The use of lasers to simulate radiation-induced transients in semiconductor devices and circuits. *IEEE Transactions on Nuclear Science*, 12(5):91–100, 1965.

[45] Y.-i. Hayashi, N. Homma, T. Sugawara, T. Mizuki, T. Aoki, and H. Sone. Non-invasive emi-based fault injection attack against cryptographic modules. In *2011 IEEE International Symposium on Electromagnetic Compatibility*, pages 763–767. IEEE, 2011.

[46] W. He, J. Breier, and S. Bhasin. Cheap and cheerful: A low-cost digital sensor for detecting laser fault injection attacks. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 27–46. Springer, 2016.

[47] W. He, J. Breier, S. Bhasin, D. Jap, H. G. Ong, and C. L. Gan. Comprehensive laser sensitivity profiling and data register bit-flips for cryptographic fault attacks in 65 nm fpga. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 47–65. Springer, 2016.

[48] X. Hou, J. Breier, D. Jap, L. Ma, S. Bhasin, and Y. Liu. Physical security of deep learning on edge devices: Comprehensive evaluation of fault injection attack vectors. *Microelectronics Reliability*, 120:114116, 2021.

[49] M. Joye and M. Tunstall. *Fault analysis in cryptography*, volume 147. Springer, 2012.

[50] D. Karaklajić, J.-M. Schmidt, and I. Verbauwhede. Hardware designer's guide to fault attacks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(12):2295–2306, 2013.

[51] M. S. Kelly and K. Mayes. High precision laser fault injection using low-cost components. In *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 219–228. IEEE, 2020.

[52] Z. Kenjar, T. Frassetto, D. Gens, M. Franz, and A.-R. Sadeghi. {V0LTpwn}: Attacking x86 processor integrity from software. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1445–1461, 2020.

[53] F. Khelil, M. Hamdi, S. Guilley, J. L. Danger, and N. Selmane. Fault analysis attack on an fpga aes implementation. In *2008 New Technologies, Mobility and Security*, pages 1–5. IEEE, 2008.

[54] C. H. Kim and J.-J. Quisquater. Faults, injection methods, and fault attacks. *IEEE Design & Test of Computers*, 24(6):544–545, 2007.

[55] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu. Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. *ACM SIGARCH Computer Architecture News*, 42(3):361–372, 2014.

[56] Y. Konishi, M. Kumanoya, H. Yamasaki, K. Dosaka, and T. Yoshihara. Analysis of coupling noise between adjacent bit lines in megabit drams. *IEEE Journal of Solid-State Circuits*, 24(1):35–42, 1989.

[57] R. Korkikian, S. Pelissier, and D. Naccache. Blind fault attack against spn ciphers. In *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 94–103. IEEE, 2014.

[58] R. Kumar, P. Jovanovic, and I. Polian. Precise fault-injections using voltage and temperature manipulation for differential cryptanalysis. In *2014 IEEE 20th International On-Line Testing Symposium (IOLTS)*, pages 43–48. IEEE, 2014.

[59] A. Kurmus, N. Ioannou, M. Neugschwandtner, N. Papandreou, and T. Parnell. From random block corruption to privilege escalation: A filesystem attack vector for rowhammer-like attacks. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.

[60] Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta. Fault sensitivity analysis. In *International workshop on cryptographic hardware and embedded systems*, pages 320–334. Springer, 2010.

[61] P. Luo, Y. Fei, L. Zhang, and A. A. Ding. Differential fault analysis of sha3-224 and sha3-256. In *2016 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 4–15. IEEE, 2016.

[62] M. Madau, M. Agoyan, J. Balasch, M. Grujić, P. Haddad, P. Maurine, V. Rožić, D. Singelée, B. Yang, and I. Verbauwhede. The impact of pulsed electromagnetic fault injection on true random number generators. In *2018 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 43–48. IEEE, 2018.

[63] J. A. Mandelman, R. H. Dennard, G. B. Bronner, J. K. DeBrosse, R. Divakaruni, Y. Li, and C. J. Radens. Challenges and future directions for the scaling of dynamic random-access memory (dram). *IBM Journal of Research and Development*, 46(2.3):187–212, 2002.

[64] S. Mangard, E. Oswald, and T. Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.

[65] N. Moro, A. Dehbaoui, K. Heydemann, B. Robisson, and E. Encrenaz. Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller. In *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 77–88. IEEE, 2013.

[66] K. Murdock, D. Oswald, F. D. Garcia, J. Van Bulck, D. Gruss, and F. Piessens. Plundervolt: Software-based fault injection attacks against intel sgx. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1466–1482. IEEE, 2020.

[67] O. Mutlu and J. S. Kim. Rowhammer: A retrospective. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(8):1555–1571, 2019.

[68] C. O'Flynn. {MIN()imum} failure:{EMFI} attacks against {USB} stacks. In *13th USENIX Workshop on Offensive Technologies (WOOT 19)*, 2019.

[69] M. Oliverio, K. Razavi, H. Bos, and C. Giuffrida. Secure page fusion with vusion: https://www. vusec. net/projects/vusion. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 531–545, 2017.

[70] R. Omarouayache, J. Raoult, S. Jarrix, L. Chusseau, and P. Maurine. Magnetic microprobe design for em fault attack. In *2013 International Symposium on Electromagnetic Compatibility*, pages 949–954. IEEE, 2013.

[71] S. Ordas, L. Guillaume-Sage, and P. Maurine. Electromagnetic fault injection: the curse of flip-flops. *Journal of Cryptographic Engineering*, 7(3):183–197, 2017.

[72] S. Patranabis, J. Breier, D. Mukhopadhyay, and S. Bhasin. One plus one is more than two: a practical combination of power and fault analysis attacks on present and present-like block ciphers. In *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 25–32. IEEE, 2017.

[73] S. Patranabis, A. Chakraborty, and D. Mukhopadhyay. Fault tolerant infective countermeasure for aes. In *International conference on security, privacy, and applied cryptography engineering*, pages 190–209. Springer, 2015.

[74] C. Patrick, B. Yuce, N. F. Ghalaty, and P. Schaumont. Lightweight fault attack resistance in software using intra-instruction redundancy. In *International Conference on Selected Areas in Cryptography*, pages 231–244. Springer, 2016.

[75] S. Pinto and N. Santos. Demystifying arm trustzone: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

[76] P. Qiu, D. Wang, Y. Lyu, and G. Qu. Voltjockey: Breaching trustzone by software-controlled voltage manipulation over multi-core frequencies. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 195–209, 2019.

[77] P. Qiu, D. Wang, Y. Lyu, R. Tian, C. Wang, and G. Qu. Voltjockey: A new dynamic voltage scaling-based fault injection attack on intel sgx. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(6):1130–1143, 2020.

[78] A. S. Rakin, Z. He, J. Li, F. Yao, C. Chakrabarti, and D. Fan. T-bfa: Targeted bit-flip adversarial weight attack. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[79] K. Razavi, B. Gras, E. Bosman, B. Preneel, C. Giuffrida, and H. Bos. Flip feng shui: Hammering a needle in the software stack. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1–18, 2016.

[80] S. Saha, A. Bag, D. Basu Roy, S. Patranabis, and D. Mukhopadhyay. Fault template attacks on block ciphers exploiting fault propagation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 612–643. Springer, 2020.

[81] L. Sauvage. Electric probes for fault injection attack. In *2013 Asia-Pacific Symposium on Electromagnetic Compatibility (APEMC)*, pages 1–4. IEEE, 2013.

[82] J.-M. Schmidt and M. Hutter. *Optical and EM fault-attacks on CRT-based RSA: Concrete results*. 2007.

[83] T. Schneider, A. Moradi, and T. Güneysu. Parti–towards combined hardware countermeasures against side-channel and fault-injection attacks. In *Annual International Cryptology Conference*, pages 302–332. Springer, 2016.

[84] M. Seaborn and T. Dullien. Exploiting the dram rowhammer bug to gain kernel privileges. *Black Hat*, 15:71, 2015.

[85] N. Selmane, S. Guilley, and J.-L. Danger. Practical setup time violation attacks on aes. In *2008 Seventh European Dependable Computing Conference*, pages 91–96. IEEE, 2008.

[86] T. Simon, L. Batina, J. Daemen, V. Grosso, P. M. C. Massolino, K. Papagiannopoulos, F. Regazzoni, and N. Samwel. Friet: an authenticated encryption scheme with built-in fault detection. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 581–611. Springer, 2020.

[87] S. P. Skorobogatov and R. J. Anderson. Optical fault induction attacks. In *International workshop on cryptographic hardware and embedded systems*, pages 2–12. Springer, 2002.

[88] J. M. Soden and R. E. Anderson. Ic failure analysis: techniques and tools for quality reliability improvement. *Proceedings of the IEEE*, 81(5):703–715, 1993.

[89] S. Tajik, H. Lohrke, F. Ganji, J.-P. Seifert, and C. Boit. Laser fault attack on physically unclonable functions. In *2015 workshop on fault diagnosis and tolerance in cryptography (FDTC)*, pages 85–96. IEEE, 2015.

[90] A. Tang, S. Sethumadhavan, and S. Stolfo. {CLKSCREW}: Exposing the perils of {Security-Oblivious} energy management. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1057–1074, 2017.

[91] A. Tatar, R. K. Konoth, E. Athanasopoulos, C. Giuffrida, H. Bos, and K. Razavi. Throwhammer: Rowhammer attacks over the network and defenses. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 213–226, 2018.

[92] N. Timmers and C. Mune. Escalating privileges in linux using voltage fault injection. In *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 1–8. IEEE, 2017.

[93] V. Van Der Veen, Y. Fratantonio, M. Lindorfer, D. Gruss, C. Maurice, G. Vigna, H. Bos, K. Razavi, and C. Giuffrida. Drammer: Deterministic rowhammer attacks on mobile platforms. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1675–1689, 2016.

[94] A. Vasselle, H. Thiebeauld, Q. Maouhoub, A. Morisset, and S. Ermeneux. Laser-induced fault injection on smartphone bypassing the secure boot. In *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 41–48. IEEE, 2017.

[95] Y. Xiao, X. Zhang, Y. Zhang, and R. Teodorescu. One bit flips, one cloud flops:{Cross-VM} row hammer attacks and privilege escalation. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 19–35, 2016.

[96] F. Zhang, X. Lou, X. Zhao, S. Bhasin, W. He, R. Ding, S. Qureshi, and K. Ren. Persistent fault analysis on block ciphers. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 150–172, 2018.

[97] L. Zussa, A. Dehbaoui, K. Tobich, J.-M. Dutertre, P. Maurine, L. Guillaume-Sage, J. Clediere, and A. Tria. Efficiency of a glitch detector against electromagnetic fault injection. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2014.