Large Laser Spots and Fault Sensitivity Analysis

Falk Schellenberg*, Markus Finkeldey†, Nils Gerhardt†, Martin Hofmann†, Amir Moradi* and Christof Paar*
*Horst Görtz Institute for IT-Security
†Photonics and Terahertz-Technology
Ruhr-Universität Bochum, Germany
{firstname.lastname}@rub.de

***Abstract*—Laser Fault Injection (LFI) is a powerful method of introducing faults into a specific area of an integrated circuit. Because the minimum spot size of the laser spot is physically bounded, many recent publications investigate down to which technology node individual transistors can be targeted. In contrast, we develop a novel attack that is applicable even when a large number of gates is affected at the smallest feature sizes. To achieve this, we adapt Fault Sensitivity Analysis to the laser setting. Such attacks require reasoning about the critical path of a combinatorial circuit and were previously only considered for clock glitches. Indeed, we show that this prerequisite is available for LFI as well. This leads to a very relaxed fault model, especially in terms of the required laser spot size. We conclude that there is no intrinsic protection for the latest technology nodes and LFI remains a serious threat for embedded devices. Experimental results are provided by targeting the combinatorial AES Sbox of an Atmel ATxmega microcontroller with an artificially large laser spot. Finally, we discuss why this attack is still applicable to the smallest structure sizes.**

***Index Terms*—fault injection, laser fault injection, fault sensitivity analysis, collision attack, AES, ATxmega**

I. Introduction

The introduction of fault injection attacks in [1] raised a large research interest in finding (mathematical) attacks on different ciphers, physical effects to introduce a fault, and countermeasures to protect a circuit against such attacks (cf. [2], [3]). The basic idea behind fault injection is to operate the Device Under Test (DUT) beyond its specification so that a computational error occurs. Depending on the appearing faults, mathematical evaluation of the obtained faulty and genuine ciphertexts might reveal the secret key. Physical possibilities to introduce a fault include clock glitches, supply voltage spikes, electromagnetic (EM) radiation, and optical fault injection. In particular Laser Fault Injection (LFI), as introduced in [4], has one important advantage. The methods previously mentioned usually affect the whole device (clock, voltage) or a very large area (EM). Instead, LFI allows targeting multiple or even single transistors with a laser beam. Because of the diffraction limit given by the wave nature of light, there is a physical limit for the spot size of the laser beam. Consequentially, many recent publications investigate down to which technology node individual transistors can be targeted without affecting the neighboring ones (cf. [5], [6], [7], [8], [9], [10]). A potential barrier might be reached at 45 nm [5].

Single-bit faults certainly enable the most sophisticated fault models and hence, fault attacks. However, the important question is whether there are (laser-based) fault attacks which do not require this precision. Indeed, we discuss in the following that hitting a few or even a very large number of transistors with a laser beam can lead to useful faults — and that there is no intrinsic protection against LFI even at very small technology nodes. To achieve this, we develop a novel attack by adapting Fault Sensitivity Analysis (FSA) [11] in its enhanced version of [12] to the LFI setting. This results in a very relaxed fault model particularly in terms of the required spot size. Large laser spots were previously only considered in [13]. Yet, the front side was targeted, which might not be applicable for integrated circuits with many metal layers and metal fill [14]. To the best of our knowledge, we are the first to consider timing violations by laser excitation.

The remainder of this paper is structured as follows. First, we briefly recapitulate the idea behind FSA and show in Sect. III that the prerequisite for the original attack is available in the LFI setting as well. Then, we practically evaluate the attack targeting the hardware AES-encryption of an Atmel ATxmega16 microcontroller in Sect. IV. Finally, we discuss the impact of the technology node in Sect. V.

II. Fault Sensitivity Analysis

FSA was originally introduced in [11] and targets the combinatorial part of an implementation. The important observation here is that the length of the critical path between two registers is dependent on the input of the circuit (cf. Fig. 1). In more detail, the accumulated propagation delay until the output of the circuit is stable depends mainly on two parameters: the active gates (different depths) and the signals applied to these gates. To exploit this behavior, the authors targeted different AES Sboxes implemented on an ASIC prototype by introducing

The work described in this paper has been supported by the German Federal Ministry of Education and Research BMBF (grant 16KIS00-15/26/27, Project PhotonFX^2).

clock glitches with increasing frequency. Faults started to occur at a certain frequency, called the critical fault injection intensity. For the attack, the authors correlated this critical frequency for multiple random plaintexts with a prediction based on the ciphertexts and a key hypothesis. Using only 50 plaintexts, the authors successfully attacked a 128-bit PPRM1 AES implementation. The downside of this approach is the mentioned prediction function which has to be generated using extensive device profiling.

As an improvement to this attack, two options to apply the Correlation-Enhanced Collision attack of [15] to FSA were discussed in [12]. Both options skip the profiling step entirely. These attacks also require that the input dependency of the critical path is similar for different instantiations of the same circuit. We omit Option 2 of [12] here since Option 1 is more similar to our approach. Option 1 captures the distribution of the resulting ciphertexts when setting the clock glitch leads to approximately 50 % faulty ciphertexts. The authors create a list $Cnt(i)$ that stores how often the faulty ciphertext i occurred. Repeating this for multiple instances of the Sbox allows finding collisions. Assuming the hypothetical difference between two bytes of the key $\Delta k = k_i \oplus k_j$ is zero, i.e., $\Delta k = 0$, the distributions are expected to be similar. Other possible hypotheses can be easily tested by rearranging one of the distributions, i.e., $Cnt'(i) = Cnt(i \oplus \Delta k_{hyp})$. For measuring the similarity of the distributions, the authors used the Pearson correlation coefficient. Thus, the most probable hypothesis is expected to show the highest correlation. Note that full control over the clock signal is not available in every setting. For example, the clock is usually generated internally for security-enhanced smartcards.

III. Laser Fault Sensitivity Analysis

We adapt the idea of FSA to the laser setting. A detailed explanation of the (physical) effects concerning LFI is available in [4], [16]. When the photons of a laser beam hit the pn-junction of a transistor, a current is created that might charge or discharge the output of the targeted gate. When targeting, e.g., Static Random Access Memory (SRAM) or flip-flops, the state might be permanently altered. However, when shooting at general combinatorial logic, the effect caused by the laser is only transient. The circuit will regain its original state depending on its input when the laser excitation is stopped. We use this transient effect as the basis for FSA as described in the following.

A. Timing Violations by Different Laser Pulse Lengths

Figure 1 depicts the general setting for (clocked) combinatorial logic. At the first rising edge of the clock signal (c_1), register A becomes transparent and applies its input to the output on its right. Then, the logic gates of the combinatorial circuit switch consecutively, each gate with a specific delay. Exactly at the next rising edge of the clock signal (c_2), the values present at the input of register B are stored. Hence, the total propagation delay through the combinatorial circuit has to be smaller than the clock period. Otherwise, faulty intermediate values might be stored in register B.

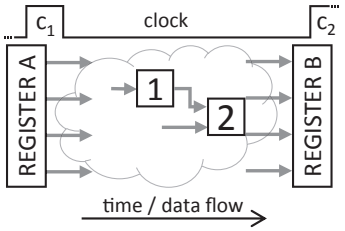


Fig. 1: General structure of clocked combinatorial logic between two registers.

Algorithm 1 Measurement phase for a single Sbox

Input: random plaintext P, number of executions N, target byte $j \in \{0, ..., 15\}$
Output: The number of faults $Cnt^j(p)$ $(p = 0, 1, .., 255)$ that occurred for plaintext byte $P^j = p$

```
Cnt^j(p) ← 0 for p = 0, 1, ..., 255
for n = 1 to N do
    run encryption with injected fault
    if fault occurred then
        Cnt^j(p) ← Cnt^j(p) + 1
    end if
end for
```

Algorithm 2 Evaluation phase

Input: Target bytes i and j, respective distributions Cnt^i and Cnt^j
Output: Most probable key difference $\Delta k = k_i \oplus k_j$

```
for 0 ≤ Δk ≤ 255 do
    Cnt'^j(a) ← Cnt^j(a ⊕ Δk), ∀a ∈ {0, ..., 255}
    Cor(Δk) = ρ(Cnt^i, Cnt'^j)   // Pearson Correlation
end for
return arg max_Δk Cor(Δk)
```

The blocks *1* and *2* in Fig. 1 represent individual gates or entire subnets. Now we target the laser beam at block *1* and force its output to some faulty value. Consequentially, block *2* has to re-evaluate and at some point in time, the faulty values will have propagated to the input of register B. Here we can observe two extrema regarding the end of the laser pulse with respect to the distance to the following clock edge (c_2). First, when the pulse ends long *after* the rising edge c_2, the faulty value will surely be latched into register B and the fault is preserved. The second case is that the laser pulse stops long *before* clock event c_2. After the laser influence, block *1* restores its original value and block *2* has enough time to re-evaluate. Then, the correct values are latched into register B and the fault is not preserved. Now we consider an offset or pulse length so

that the laser ends within this interval. Then, we expect to see multiple different faulty values in register B. With increasing pulse length, an increasing number of gates in the path(s) from the fault to the register will affect the result. Note that the above might be exploitable using Fault Intensity Analysis [17] as well. There, it is assumed that for increasing fault intensity the number of affected bits increases likewise. The characterization of our DUT confirms this behavior. Regardless, it is not required for the attack to succeed.

B. Attack Strategy

From the considerations above, we conclude that LFI might provide an identical basis for FSA where in the original works clock glitches have been applied. Yet, we slightly modify the attack strategy [12]. Instead of the last round, we target SubBytes in the first round of AES. Thus, we introduce faults during the computation of $\mathrm{Sbox}(p \oplus k)$ for plaintext byte p and key byte k. We assume a serialized implementation. Hence, we repeat the following step for each Sbox instance. For random plaintexts, we increase the length of the laser pulse by small steps until a certain percentage of faults appears and then we run N trials. We create a list $Cnt(p)$ that stores the number of faults that occurred for plaintext byte p. Note that by targeting the first round, we *do not* require the exact value of the genuine or the faulty ciphertext but solely the plaintext and knowledge of whether a fault occurred or not. For evaluation, we use Pearson correlation to find collisions as described in [12]. Likewise, the correct difference $\Delta k = k_i \oplus k_j$ is expected to show the highest correlation. For completeness, we provide the pseudocode of the measurement phase in Alg. 1 and of the evaluation in Alg. 2.

IV. Practical Evaluation

In the following, we present a practical evaluation of the proposed attack targeting the AES co-processor of an ATxmega16. After describing the experimental setup, we show that we indeed can measure individual timings using LFI. Finally, we run the attack and successfully recover the correct key differences.

A. Experimental Setup

We conducted our experiments using an Atmel ATxmega16A4U with a minimal feature size of around 250 nm [18]. We chose this microcontroller as it allows easy access to a fully controllable real-world hardware implementation of AES. Note that an identical ATxmega16 was found to be vulnerable when targeting its flip-flops with LFI [18]. The AES core is loosely attached by status and control registers to the CPU (as opposed to round instructions). Thus, we can be certain that any obtained faults actually originated from the AES and not from other parts of the circuit like instruction registers, etc. A single AES encryption requires 375 clock cycles. We assume that a highly serialized implementation is used and most importantly, a serialized SubBytes operation as well. Figure 2 shows the silicon die, captured from the backside using NIR illumination. The area of the Sbox implementation is marked by a rectangle. The exact location and its dimension were found earlier by an exhaustive search spatially and for every input. Gates related to the Sbox were found to cover an area of $230 \times 310\,\mu\mathrm{m}^2$.

The backside of the DUT was thinned to approximately 20 µm remaining silicon substrate. The supply voltage of the DUT was set to 1.6 V and the clock was provided externally at 2 MHz. We used a microscope by Opto GmbH built for LFI, slightly modified to match our stability and throughput requirements. The laser was focused through a Mitutoyo Plan Apo NIR (Numerical Aperture (NA) 0.26, magnification 10x) objective. Two 975 nm single mode fiber-coupled on-demand diode laser modules from AL-PhANOV were used. Both were set to maximum output and were focused on the same spot to maximize the energy density. We measured a laser peak power of 0.52 W. Because of the overly long pulse widths (cf. Fig. 3), the pulse energy is not relevant. The timing was controlled by a Stanford Research Systems DG645 programmable delay generator based on a trigger signal at the beginning of the encryption.

First, we established the optimal focal plane by gradually decreasing the energy to a minimum. During this process, we adjusted the location of the spot spatially and axially in such a way that faults were still observable. The minimum spot size for an $\mathrm{NA} = 0.26$ is calculated as $\frac{1.22 \times \lambda}{NA} = 4.5\,\mu\mathrm{m}$ at a wavelength of $\lambda = 975\,\mathrm{nm}$ (diffraction limit by Abbe). We measured the spot size using an Ophir Spiricon SP620U beam profiling camera with a pixel pitch of 4.4 µm. Indeed, the minimal measured spot size was a single pixel. For our experiments, we intentionally changed the focal plane to simulate smaller technology nodes. The resulting spot size using an offset of 85 µm was measured to be approximately 45 µm (Gaussian spot, intensity above 10 %.), i.e., larger by a factor of 10. The combinatorial logic of the ATxmega is made up of CMOS lanes of 12 µm [18]. Thus, the area illuminated by the laser beam contains more than three entire lanes in width and a multitude of transistors with respect to the layout given in [18].

Figure 3 depicts the laser pulse with respect to other relevant signals. The digital signals were scaled and shifted for better visibility as their absolute value is of no relevance. The black and gray signals represent the current flowing through the device measured over a shunt resistor. The laser was not powered for the gray signal. When powering the laser (black), the effect on the measured current can be clearly identified (cf. Optical Beam Induced Current (OBIC) [18]). The blue signal represents the trigger sent from the delay generator to the laser diode module. The laser pulse in purple was measured through a photo-diode in the optical path (converted to a voltage by a shunt and amplified using a Langer PA303 amplifier).

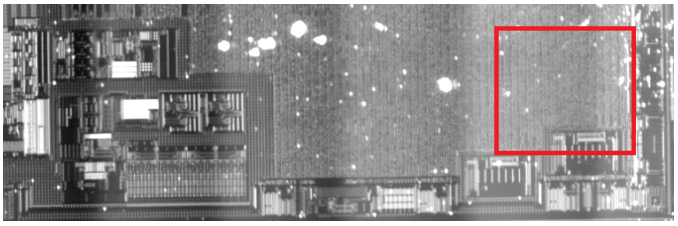


Fig. 2: ATxmega16A4U backside image using NIR illumination, bottom-right corner (mirrored as seen from the front side). The rectangle marks the position of the combinatorial Sbox implementation.

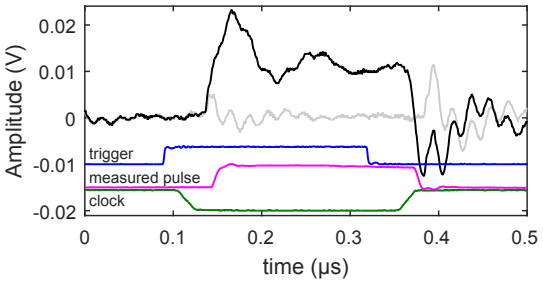


Fig. 3: Timing diagram of the fault injection. Black: current through the device (laser on), gray: current (laser off), green: clock signal, blue: pulse to the laser diodes, purple: measured pulse (clock, trigger, and measured pulse are scaled for visibility).

In fact, using this exact configuration, we did not obtain any faults. However, increasing the length of the pulse by less than 10 ns already resulted in stable faults, i.e., the faulty values did not change with increasing the length. It might seem confusing that the measured pulse stops slightly after the rising edge of the clock. However, this is due to the positions in which the respective signals were measured. We can only measure the clock signal at the clock generator and *not* at the exact time it arrives at the respective register. However, assuming that the signal from the photo-diode and the current of the DUT have a similar propagation delay, we can observe the following. First, the measured pulse ends right before the foot of the peak of the gray signal around 0.4 µs. Further, the current caused by the laser (visible by the black signal) ends right before the same respective peak of the power consumption trace. Note that the laser output is stable long before the pulse ends. Thus, we do not change the rate of photons affecting the device but solely the time when the effect stops.

B. Measuring Individual Timings

Note that the following is not required for the attack but to support the assumption of measurable input dependencies using LFI made in Sect. III. We ran multiple tests for different inputs to an Sbox while increasing the pulse length in steps of 5 ps in the interval mentioned above. We have no information about the delay of individual

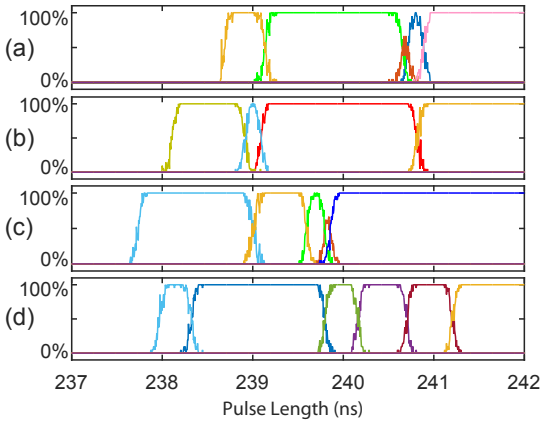


Fig. 4: Percentage of different faulty values at the output of the Sbox for increasing laser pulse length, four exemplary chosen inputs. Colors represent different faulty values at the output. The colors do not match for different inputs, i.e., all obtained faults are unique.

gates of the ATxmega processor. However, a propagation delay of 41 ps for a single inverter (ring oscillator speed per gate) in a 250 nm technology is mentioned in [19]. Thus, we assume to observe changing values in this region as well. Figure 4 depicts the obtained faulty behavior for four different (but fixed) inputs to the Sbox while increasing the pulse length. We shot 20 times for each delay value; the y-axis represents the percentage a certain faulty value appeared on the output. Considering for example plot (a), we injected no fault until a delay of 238.5 ns (measured from the start of the pulse). After around 238.5 ns, we started to obtain a certain fault at the output and for a slightly increased delay, the fault appeared in each of the 20 tests. Continuing to increase the delay, for a delay of 239.0 ns, a different faulty value appeared at the output and remained stable until approx. 240.05 ns. Note that we focused on the part with changing faults, i.e., all the faults were stable after 242 ns. For the attack, we especially require that faulty outputs start to appear for different inputs at different pulse lengths. Indeed, we can observe this effect in Fig. 4, revealing differences up to 1 ns.

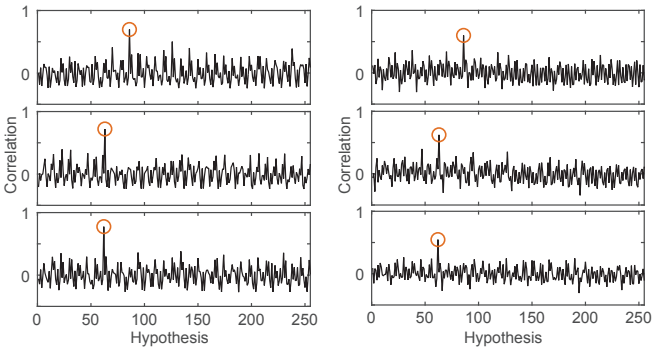
The ATxmega stores the final round key in a dedicated register, e.g., so that decryption can be performed using this key and a flag that runs the key schedule in reverse. We used this to confirm that the gates related to the key schedule were unaltered. Since we know the key used for the encryption, we can use the faulty ciphertext to calculate backwards in order to investigate the faults (cf. Fig. 4). Indeed, for every test, only the output of the targeted Sbox operation in the correct cipher round was altered.

C. Attack Results

We performed the attack outlined in Sect. III choosing the pulse length in such way that 20 %, 50 %, and 80 % of the inputs led to faulty outputs. We targeted an arbi-

trarily chosen area within the borders of the combinatorial Sbox. Since the ATxmega processes each byte of the state consecutively, we repeated the measurement for each byte by delaying the start of the pulse by one clock cycle, i.e., 500 ns. We found that the ATxmega processes the bytes in the order created after the ShiftRows operation. Thus, the hypotheses should consider the following order: $(0, 5, 10, 15, 4, 9, 14, 3, 8, 13, 2, 7, 12, 1, 6, 11)$.

Figure 5 depicts the results of the correlation collision [15] for the first three Δk using $N = 1000$ samples. For Fig. 5(a), a delay value was chosen so that approximately 20 % of the random input resulted in a fault (i.e., 200 faults out of the 1000 samples). For each target Δk (and all 12 remaining other ones), the peak corresponding to the correct hypothesis is clearly distinguishable. In Fig. 5(b), the results for 50% fault occurrence are depicted. The attack succeeds likewise, although with a smaller correlation.



(a) 20%, $N = 1000$ (b) 50%, $N = 1000$

(c) 80%, $N = 1000$ (d) 80%, $N = 5000$

Fig. 5: Correlation collision for different percentages and N, top to bottom: $\Delta k_{0,1}$, $\Delta k_{0,2}$, $\Delta k_{0,3}$. Correct Δk marked with a circle.

We ran the attack again using a pulse length long enough that all faults were stable, i.e., the laser pulse ends after the rising edge of the clock signal. This way, we obtained approximately 80 % faults, meaning that not the whole Sbox is affected by the laser beam. Our results show that the attack fails for the same amount of measurements (Fig. 5(c)). When using $N = 5000$ measurements, the correct hypotheses showed the highest correlation (cf. Fig. 5(d)). However, the peaks are again not easily distinguishable.

V. Discussion

From Sect. III, we observe that it is not critical which part of the Sbox is affected nor that we understand which fault was exactly injected. We only have to obtain *some* faulty behavior. Further and most importantly, the spot size (or the number of affected gates, respectively) is not critical either. Of course, assuming that all faults are stable and the spot size is so large that we obtain a fault for every possible input, the attack clearly fails. However, we solved this issue by introducing variable pulse lengths, as shown above. For example, when targeting block *1* in Fig. 1, it is not relevant whether we hit block *2* as well. When the laser pulse ends, block *2* has to reevaluate anyway since its input was changed by block *1*. Transferring this to the complete Sbox, only the affected gate with the longest propagation path will determine the success of the attack.

The Sbox of the ATxmega covers an area of $230 \times 310\,\mu m^2$ at a technology node of 250 nm. Although being a very rough estimation, transferring this value to, e.g., 11 nm, leads to an area of $10 \times 13\,\mu m^2$. This still provides enough space to target the Sbox with LFI without hitting unrelated logic. For example, applying the Abbe diffraction limit to high-resolution long working distance NIR objectives results in a minimum spot size of 1.7 µm for a commercially available NA of 0.7. Note that scaling down our artificially large spot to 11 nm likewise leads to a spot size of 1.98 µm.

Since we are trading spatial accuracy with timing precision, it should be noted that current laser systems certainly make no limits in terms of pulse length and jitter. Even considering clock frequencies above 1 GHz or clock periods below 1 ns, the clock jitter has to scale with the frequency. Current research in laser technology considers jitter in the atto-second range for femto-second lasers [20]. In contrast, [19] mentions a delay value of 4.98 ps for a 40 nm process.

VI. Conclusion and Future Work

By adapting Fault Sensitivity Analysis to the LFI setting, we inherit a convenient fault model. For example, we allow random plaintexts and do not require genuine or faulty ciphertexts. As opposed to classical Differential Fault Analysis, we solely use the information whether a fault occurred or not. Compared to the original approaches, we trade clock glitches for laser fault injection and do not require control over the clock signal anymore. We show that it is indeed possible to exploit input-dependent timing violations using fine-adjusted laser pulse lengths. Most importantly, we obtain very loose requirements in terms of the laser spot size. The attack still succeeds even if the laser spot is so large that every input to the combinatorial logic is affected. To support our claims, we provided experimental results targeting an

AES hardware implementation with an artificially large spot and successfully extracted the secret key. Theoretically scaling the used parameters to the latest technology nodes still provides plenty of space to perform the attack. Ongoing research on down to which feature size single bits (or transistors) can be individually addressed with LFI might reach a limit soon (cf. 45 nm [5]). In contrast, we stress that still, such feature sizes themselves are unable to offer inherent security and have to be protected against fault attacks and especially LFI. Note that the proposed attack is not limited to (AES-)Sboxes but rather applies to combinatorial circuits in general.

Replacing clock glitches by laser fault injection as described in this work might be a promising aspect for future research. This offers to introduce timing violations into a specific area of the chip opposed to affecting the whole area. More specifically, this holds especially for very high frequencies where clock glitches might not reach the target due to board and on-chip capacitances acting as a low-pass filter.

References

[1] E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," in *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, 1997, pp. 513–525. [Online]. Available: http://dx.doi.org/10.1007/BFb0052259

[2] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The Sorcerer's Apprentice Guide to Fault Attacks," *IACR Cryptology ePrint Archive*, vol. 2004, p. 100, 2004. [Online]. Available: http://eprint.iacr.org/2004/100

[3] D. Karaklajic, J. Schmidt, and I. Verbauwhede, "Hardware Designer's Guide to Fault Attacks," *IEEE Trans. VLSI Syst.*, vol. 21, no. 12, pp. 2295–2306, 2013. [Online]. Available: http://dx.doi.org/10.1109/TVLSI.2012.2231707

[4] S. P. Skorobogatov and R. J. Anderson, "Optical Fault Induction Attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, vol. 2523, pp. 2–12. [Online]. Available: http://dx.doi.org/10.1007/3-540-36400-5_2

[5] B. Selmke, S. Brummer, J. Heyszl, and G. Sigl, "Precise Laser Fault injections into 90 nm and 45 nm SRAM-cells," in *Smart Card Research and Advanced Applications (CARDIS) 2015*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015.

[6] A. P. Mirbaha, J.-M. Dutertre, A.-L. Ribotta, M. Agoyan, A. Tria, and D. Naccache, "Single-Bit DFA Using Multiple-Byte Laser Fault Injection." in *10th IEEE International Conference on Technologies for Homeland Security*, Boston, United States, Nov. 2010. [Online]. Available: http://hal-emse.ccsd.cnrs.fr/emse-00552195

[7] M. Agoyan, J. Dutertre, A. Mirbaha, D. Naccache, A. Ribotta, and A. Tria, "How to flip a bit?" in *16th IEEE International On-Line Testing Symposium (IOLTS 2010), 5-7 July, 2010, Corfu, Greece*, 2010, pp. 235–239. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/IOLTS.2010.5560194

[8] F. Courbon, P. Loubet-Moundi, J. J. A. Fournier, and A. Tria, "Adjusting Laser Injections for Fully Controlled Faults," in *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, 2014, pp. 229–242. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10175-0_16

[9] C. Roscian, A. Sarafianos, J. Dutertre, and A. Tria, "Fault Model Analysis of Laser-Induced Faults in SRAM Memory Cells," in *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013*, 2013, pp. 89–98. [Online]. Available: http://dx.doi.org/10.1109/FDTC.2013.17

[10] F. Courbon, P. Loubet-Moundi, J. J. A. Fournier, and A. Tria, "Increasing the efficiency of laser fault injections using fast gate level reverse engineering," in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014, Arlington, VA, USA, May 6-7, 2014*, 2014, pp. 60–63. [Online]. Available: http://dx.doi.org/10.1109/HST.2014.6855569

[11] Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta, "Fault Sensitivity Analysis," in *Cryptographic Hardware and Embedded Systems, CHES 2010*, ser. Lecture Notes in Computer Science, S. Mangard and F.-X. Standaert, Eds. Springer Berlin Heidelberg, 2010, vol. 6225, pp. 320–334. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15031-9_22

[12] A. Moradi, O. Mischke, C. Paar, Y. Li, K. Ohta, and K. Sakiyama, "On the Power of Fault Sensitivity Analysis and Collision Side-Channel Attacks in a Combined Setting," in *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, 2011, pp. 292–311. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23951-9_20

[13] C. Roscian, J. Dutertre, and A. Tria, "Frontside laser fault injection on cryptosystems - Application to the AES' last round -," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2013, Austin, TX, USA, June 2-3, 2013*, 2013, pp. 119–124. [Online]. Available: http://dx.doi.org/10.1109/HST.2013.6581576

[14] J. G. J. van Woudenberg, M. F. Witteman, and F. Menarini, "Practical Optical Fault Injection on Secure Microcontrollers," in *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2011, Tokyo, Japan, September 29, 2011*, 2011, pp. 91–99. [Online]. Available: http://dx.doi.org/10.1109/FDTC.2011.12

[15] A. Moradi, O. Mischke, and T. Eisenbarth, "Correlation-Enhanced Power Analysis Collision Attack," in *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, 2010, pp. 125–139. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15031-9_9

[16] D. H. Habing, "The use of lasers to simulate radiation-induced transients in semiconductor devices and circuits," *Nuclear Science, IEEE Transactions on*, vol. 12, no. 5, pp. 91–100, Oct 1965.

[17] N. F. Ghalaty, B. Yuce, M. M. I. Taha, and P. Schaumont, "Differential fault intensity analysis," in *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014, Busan, South Korea, September 23, 2014*, 2014, pp. 49–58. [Online]. Available: http://dx.doi.org/10.1109/FDTC.2014.15

[18] F. Schellenberg, M. Finkeldey, B. Richter, M. Schäpers, N. Gerhardt, M. Hofmann, and C. Paar, "On the Complexity Reduction of Laser Fault Injection Campaigns using OBIC Measurements," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2015 Workshop on*, Sept 2015.

[19] Europractice-IC, "TSMC 0.25 um technology overview (MPW)," accessed 30.10.2015, available at http://www.europractice-ic.com/technologies_TSMC.php?tech_id=025um.

[20] A. J. Benedick, J. G. Fujimoto, and F. X. Kartner, "Optical flywheels with attosecond jitter," *Nature Photonics*, vol. 6, no. 2, pp. 97–100, 2012.