How to Flip a Bit?

Michel Agoyan, Jean-Max Dutertre, Amir-Pasha Mirbaha, David Naccache, Anne-Lise Ribotta, Assia Tria

On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International
http://dx.doi.org/10.1109/IOLTS.2010.5560194

How to Flip a Bit?

Michel Agoyan*, Jean-Max Dutertre†, Amir-Pasha Mirbaha†, David Naccache‡, Anne-Lise Ribotta† and Assia Tria*
*†Département Systèmes et Architectures Sécurisées (SAS)
*CEA-LETI, Gardanne, France
{michel.agoyan, assia.tria}@cea.fr
†École Nationale Supérieure des Mines de Saint-Étienne (ENSMSE), Gardanne, France
{dutertre, mirbaha, ribotta}@emse.fr
‡Équipe de cryptographie, École normale supérieure, Paris, France
david.naccache@ens.fr

***Abstract*—This note describes laser fault experiments on an 8-bit 0.35μm microcontroller with no countermeasures. We show that *reproducible* single-bit faults, often considered unfeasible, can be obtained by careful beam-size and shot-instant tuning.**

I. Introduction

Fault attacks consist in using hardware malfunction to infer secrets from the target's faulty outputs. Within fault attacks, Differential Fault Analysis [2] (DFA) is a particular analysis technique exploiting differences between correct and faulty outputs. We refer the reader to [14] for more information on fault injection techniques.

This note describes laser faults experiments on an 8-bit 0.35μm RISC microcontroller with no countermeasures. We show that *reproducible* single-bit faults, often considered unfeasible, can be obtained by careful beam-size and shot-instant tuning. Moreover, we obtain such faults even when the beam's impact area exceeds a single SRAM cell. This underlines the need to protect small data objects, such as pointers, counters or flags, against "surgical" faults targeting a single-bit on a specific byte in memory and nothing else.

II. The Advanced Encryption Standard

We assume that the reader is familiar with the AES [10] that we recall here for the ease of reference.

The AES-128 (hereafter AES) encrypts 128-bit plaintexts into 128-bit ciphertexts using a 128-bit key K. The algorithm performs 10 rounds (after a short initial round) and consists of two separated processes: a key schedule that derives round keys and the encryption routine itself. During decryption key schedule is reversed and encryption is replaced by a very similar decryption process.

The initial round uses $K_0 = K$ as a round key; for all subsequent rounds, new round keys K_i are calculated from their predecessors K_{i-1}. Figure 1 illustrates the AES' structure.

In most implementations the K_is are computed and stored in memory before encryption starts. Encryption treats the 16-byte plaintext M as a 4×4 byte matrix. Each round, except the initial and the final, includes four steps: A substitution of the matrix's contents using a lookup table (`SubBytes`), a rotation

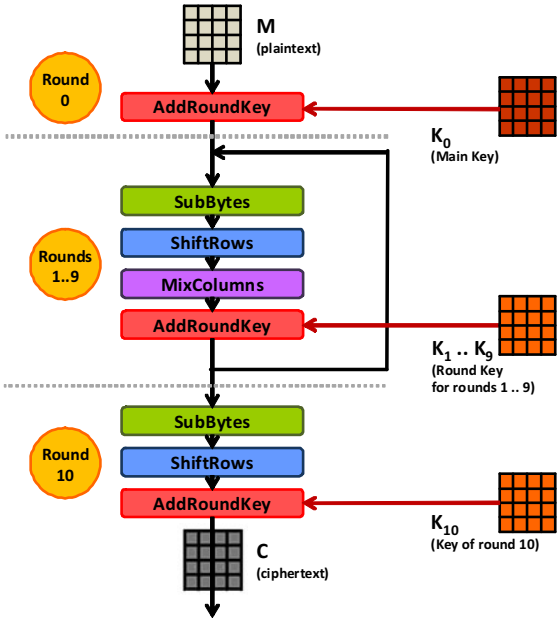

Fig. 1. The AES - General Outline.

of the matrix's rows (`ShiftRows`) and a linear transform in GF(2^8) (`MixColumns`) combining each matrix element with other column elements weighted by different coefficients (1, 2 or 3). At the end of each round, K_i is xored with `MixColumns`'s result (an operation called `AddRoundKey`).

III. Laser Fault Injection

Laser (Light Amplification by Stimulated Emission of Radiation) is a stimulated-emission electromagnetic radiation in the visible or the invisible domain. Laser light is monochromatic, unidirectional, coherent and artificial (*i.e.* laser does not spontaneously exist in nature). Laser light can be generated as a beam of very small diameter (a few μm). The beam can pass through various material obstacles before impacting a target during a very short duration.

Laser impacts on electronic circuits are known to alter functioning. Current chip manufacturing technologies are in

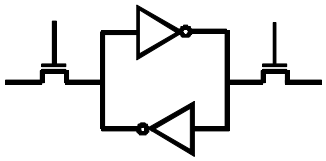

Fig. 2. Architecture of a typical SRAM cell.

the nanometers range. This, and the laser's brief and precise reaction time, makes laser a particularly suitable fault injection means.

A. Photoelectrical Effects of Laser on Silicon

SRAM (Static Random Access Memory) laser exposure is known to cause bit-flips [13], [6], [1], [5], a phenomenon called *Single Event Upset* (SEU). By tuning the beam's energy level below a destructive threshold, the target will not suffer any permanent damage.

A conventional one-bit SRAM cell (Figure 2) is made of two cross-coupled inverters. Every cell has two additional transistors controlling the cell's content access during write and read. As every inverter is made of two transistors, an SRAM cell contains six MOS.

In each cell, the states of four transistors encode the stored value. By design, the cell admits only two stable states: a "0" or a "1". In each stable state, two transistors are at an ON state and two others are OFF.

If a laser beam hits the drain/bulk reversed-biased PN junctions of a blocked transistor, the beam's energy may create pairs of electrons as the beam passes through the silicon. The charge carriers induced in the collection volume of the drain-substrate junction of the blocked transistor are collected and create a transient current that inverts logically the inverter's output voltage. This voltage inversion is in turn applied to the second inverter that switches to its opposite state: all in all, a bit flip happens [6], [1].

From the opponent's perspective, an additional advantage of laser fault injection is *reproducibility*. Identical faults can be repeated by carefully tuning the laser's parameters and the target's operating conditions.

B. Different Parameters in a Fault Attack by Laser

In a laser attack, the opponent usually controls the beam's diameter, wavelength, amount of emitted energy, impact coordinates (attacked circuit part) and the exposure's duration. Sometimes, the opponent may also control the impact's moment[1], the target's clock frequency, V_{cc} and temperature. Finally, laser attacks may indifferently target the chip's front side or back side.

However, the chip's front and back sides have different characteristics when exposed to a laser beam:

[1] *i.e.* the impact's synchrony with a given clock cycle of the target.

1) Front side attacks: are particularly suited to green wavelength ($\sim$ 532nm). The visibility of chips components makes positioning very easy in comparison to backside attacks. But because of the metallic interconnects' reflective effect, it is difficult to target a component with enough accuracy. In addition, progress in manufacturing technologies results in both a proliferation of metal interconnects and much smaller chips. All in all, it becomes increasingly difficult to hit a target area.

2) Backside attacks: are more successful at the infrared wavelength ($\sim$ 1064nm) as the laser needs to deeply enter the silicon. Positioning is more difficult for lack of visibility. Nonetheless, backside attacks allow to circumvent the reflective problem of metallic surfaces.

IV. Giraud's Bit DFA

Differential Fault Analysis [2] (DFA) is an analysis technique exploiting differences between correct and faulty outputs. Several *byte-level* and *bit-level* AES DFA variants exist (*e.g.* [11], [8], [9], [7], [3]). Given the dependence of these attacks on precise "surgical" fault injection, the feasibility of bit/byte-level DFA remained somewhat unclear.

[8] describes a bit-level and a byte-level DFA on AES. The bit-level attack requires the injection of a single-bit fault into a specific byte of the temporary ciphertext result of the penultimate round (M_9) (*e.g.* to inject such a fault into the 9-th round `AddRoundKey` or into the temporary ciphertext result just before the `SubBytes` input to the 10-th round).

To discover one byte of K_{10}, the attack requires to repeat a single-bit fault for at least three different plaintexts. The three faulty results are then compared to their corresponding correct ciphertexts to infer key information. We show that this attack can be implemented, even when the laser spot is wider than the SRAM's cell.

During normal processing, the value of each ciphertext (C) cell is calculated by xoring a corresponding K_{10} value with a temporary value resulting from the application of `SubBytes` (`SB`) and `ShiftRows` (`SR`) to (M_9):

$$C = \texttt{SR}[\texttt{SB}(M_9)] \oplus K_{10} \tag{1}$$

As shown in Figure 3, upon single-byte fault injection in K_9 (regardless the number of faulty bits) the faulty message will feature only one faulty byte that will leak information on one byte of K_{10}. Figure 4 shows the consequences of an injected fault in K_9 throughout the 9-th and 10-th rounds.

For the sake of clarity, we consider all subsequent equations *bytewise* thereby abstracting away `ShiftRowss` that do not affect individual byte values. Thus, (1) becomes (2):

$$C = \texttt{SB}(M_9) \oplus K_{10} \tag{2}$$

By considering the injection of the single-bit fault e on the 10-th round `SubBytes` input, the faulty ciphertext (D) can be expressed as (3):

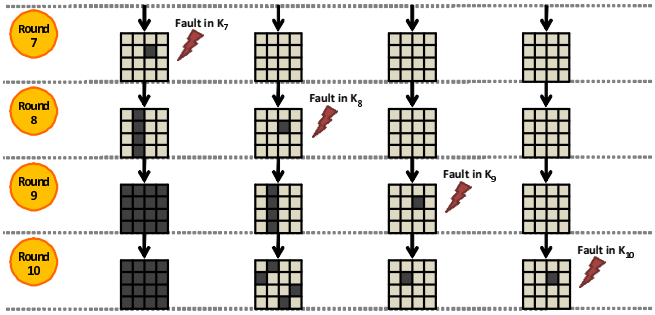

Fig. 3. Effects of one faulty round key byte at different rounds.

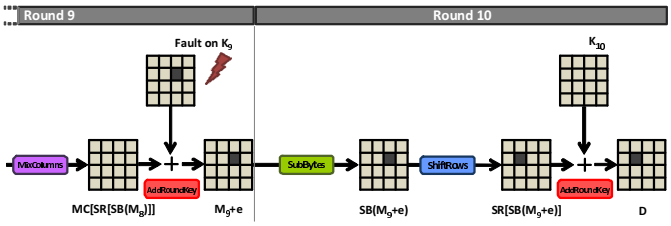

Fig. 4. Giraud's bit DFA.

$$D = \mathtt{SB}(M_9 \oplus e) \oplus K_{10} \tag{3}$$

[8] observes that a xor between a faulty and a correct ciphertext reveals a difference ($\Delta = C \oplus D$) corresponding to a set of hypotheses on the corresponding M_9 byte value before the attack, and on the injected single-bit fault e.

$$\Delta = \mathtt{SB}(M_9 \oplus e) \oplus \mathtt{SB}(M_9) \tag{4}$$

(4) will yield a set of hypotheses on possible M_9 and e value-pairs. Using (5), a corresponding K_{10} value can be replaced for each pair of (M_9, e) values.

$$K_{10} = \mathtt{SB}(M_9 \oplus e) \oplus D \tag{5}$$

By repeating the fault injection for at least three different plaintexts, the opponent creates three sets of hypotheses on the corresponding K_{10} byte value. Then, sets are intersected to spot the single hypothesis that reveals one K_{10} byte. With a probability of about 97%, three plaintexts suffice to discover a byte of K_{10} [8]. Otherwise, the opponent iterates the process for more plaintexts to until the sets' intersection reaches a singleton. After finishing this operation for one byte of K_{10}, the procedure is repeated to discover K_{10}'s remaining bytes. Finally, $K = K_0$ is inferred by reversing the key schedule.

V. Practical Single-Bit Fault Injection

Outline: After chip decapsulation and a mapping of the chip's components, we selected a large target area, given our knowledge of the implementation. Using automated search on the chip's front-side, we modified the impact's coordinates, the beam parameters and timing until a single bit fault was obtained. Finally, Giraud's bit DFA was performed.

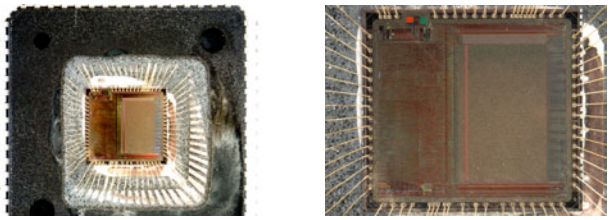

Fig. 5. Decapsulated chip (SRAM is on the left middle and bottom side).

The target is an 8-bit 0.35 μm 16 MHz RISC microcontroller with an integrated 4KB SRAM and no countermeasures. The device runs SOSSE (Simple Operating System for Smartcard Education [4]) to which we added some commands, most notably for feeding-in cleartexts and retrieving ciphertexts. K was embedded in the code. As encryption starts, the K_is are derived and stored in SRAM. The laser, shown in Figures 10 and 11, is equipped with a YAG laser emitter in three different wavelengths: green, infrared and ultraviolet.

The spot's diameter can be set between 0 and 2500μm. As the beam passes through a lens, it gets reduced by the lens' zoom factor and loses a big part of its energy. Our experiments were conducted with a 20× Mitutoyo lens, a green[2] beam of $\varnothing 4\mu$m and $\simeq 15pJ$ per shot[3]. The circuit is installed on a programmable Prior Scientific X-Y positioning table[4]. The X-Y table, card reader, laser and an FPGA trigger board, were connected via RS-232 to a control PC. The FPGA trigger board receives an activation signal from the reader and sends a trigger signal to the laser after a delay defined by the control PC.

Experiments were conducted in ambient temperature and at $V_{cc} = 5V$. These parameters are within the device's normal operating conditions $2.7V \leq V_{cc} \leq 5.5V$.

The chip was decapsulated by chemical etching using a Nisene JetEtch automated acid decapsulator. The decapsulator can be programmed for the chemical opening of different chip types using different ratios of nitric acid (HNO_3) and sulfuric acid (H_2SO_4), at a desired temperature and during a specified time. For opening our chip, we used only nitric acid at 80°C for 40 seconds. The etched chip (Figure 5) successfully passed functional tests before and during fault injection.

As it is very difficult to target the chip's (ALU) (Arithmetic Logic Unit) during a very specific instant between the end of 9-th round and the beginning of the 10-th round, we decided to target K_9.

Finding the SRAM area containing K_9 and properly tuning the laser's parameters is very time consuming. The number of faults in C, their position and their contents indicate which round key has been hit. `MixColumns` will amplify any single-bit/byte fault occurring in any K_i preceding K_9 and result in

[2] 532nm wavelength.
[3] At the laser source emitter, before passing through the lens.
[4] Motorized stepper stage for upright microscopes with 0,1 μm steps.

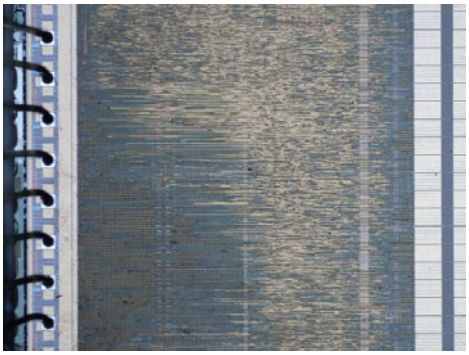

Fig. 6. Decapsulated chip (closeup on SRAM).

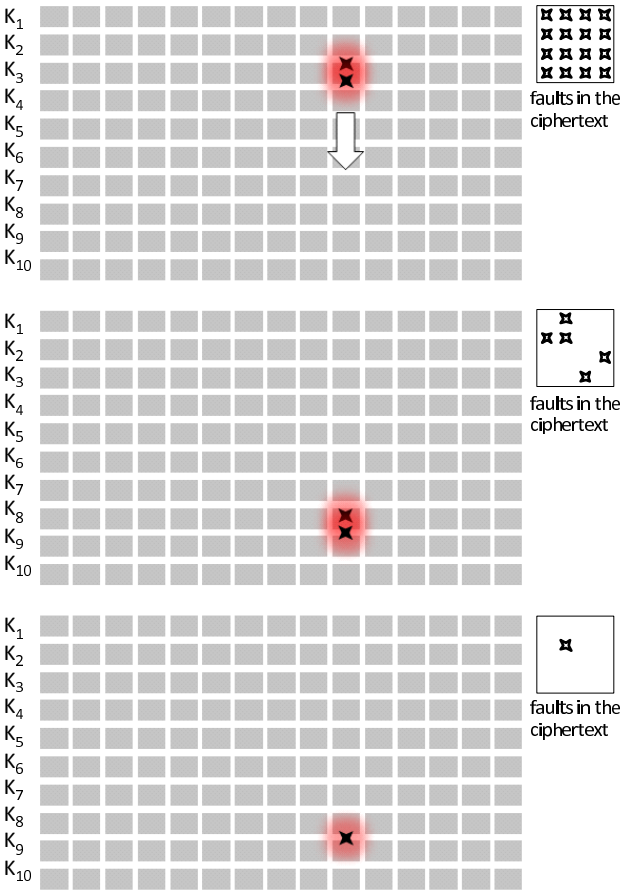

Fig. 7. Exploration process.

a multi-byte or a fully faulty ciphertext (we call such a bad event an *"early fault"*). As shown in Figure 3, a single-bit/byte fault on K_8 changes 4 bytes and on any previous K results in a completely faulty ciphertext while a fault in K_9 or K_{10} changes only one byte. However, injected faults are not always limited to a single byte and/or to a single K_i. When more than 3 ciphertext bytes are faulty, it is difficult to determine if the observed result is due to an early fault or to several faults in K_9 and K_{10} (Table I).

Figure 8 compares a 1μm laser spot and SRAM cells in

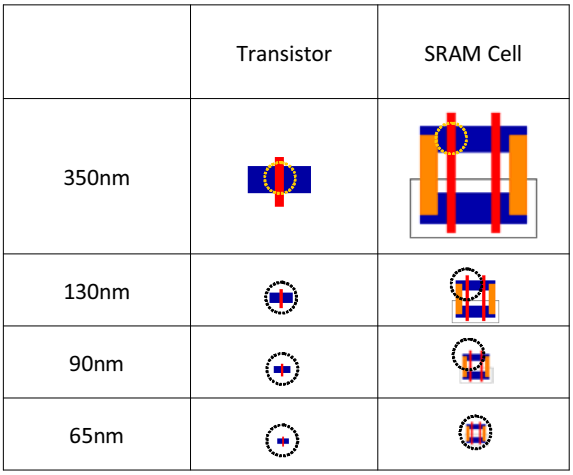

Fig. 8. 1μm laser spot (dotted circle) *vs.* technology sizes [12].

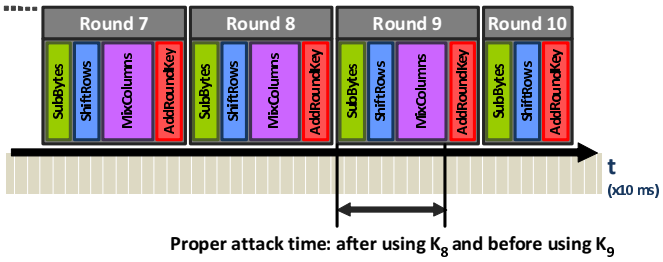

Fig. 9. Attack's timing.

TABLE I
POTENTIAL FAULTY K_iS AS FUNCTION OF OBSERVED FAULTY CIPHERTEXT BYTES.

number of faulty C bytes	potential faulty round keys			
	K_{10}	K_9	K_8	previous round keys
$1, 2, 3$	✓	✓		
$4, \ldots, 15$	✓	✓	✓	
16	✓	✓	✓	✓

different technology sizes. As technology advances, transistor density per μm grows. With several transistors are packed into 1μm areas, single-bit fault injection will require much more precise equipment and are likely to become unfeasible using cheap lasers.

Despite fine-grained energy and spatial control we detected faults in keys neighboring K_9. To overcome this problem, we isolated K_{10} from faults and restricted the shot to a 100μs interval between the use of K_8 and K_9 (Figure 9).

Figure 7 shows how we could confine faults to a single-bit of K_9. When physically more than one single-bit faulty byte existed, we could logically obtain a single-bit fault by controlling the laser's shooting time. Figure 7 is just a model of the real SRAM (Figure 6) to describe our technique and does not correspond to real address allocation. We could successfully inject a single-bit fault into each of the 16 bytes of K_9 and iterate the process for 4 different texts. This sufficed

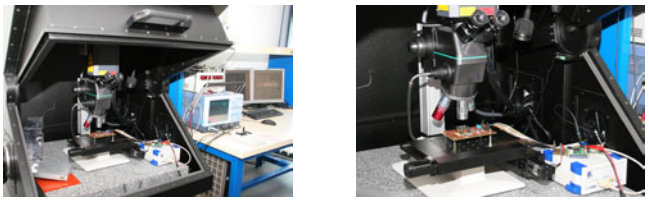

Fig. 10. Laser and target (general overview).

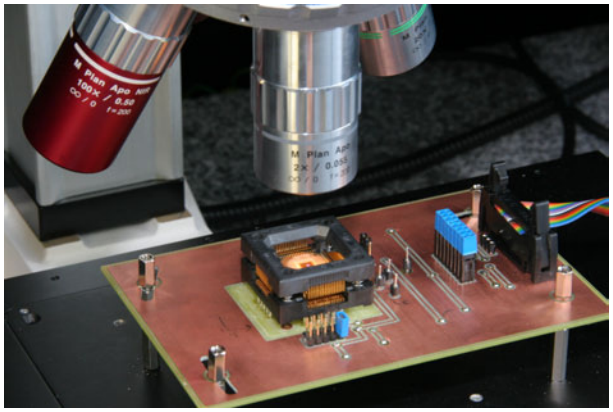

Fig. 11. Laser and target (closeup).

to succeed Giraud's bit DFA.

As shown in the topmost part of Figure 7, we searched K_9's precise storage area by monitoring the number and the type of faults in the ciphertext. Then (middle part of Figure 7), by a precise beam localization, we managed to inject faults only into K_9. This, however, did not turn out to be fully deterministic as sometimes we would also inflict faults to previous round keys. At that point (lowermost part of Figure 7), by fine-tuning spatial and temporal beam localization (just after the use of K_8), we managed to restrict the injected faults only to K_9. This is the exact assumption of Giraud's scenario for single-bit fault injection.

VI. Conclusion

We implemented single-bit Giraud's attack [8] using laser fault injection. Whilst this is not the most effective fault attack on AES, this scenario is usually regarded as the *most difficult* as it requires to limit the attack to *one single bit*. This is much more stringent that most other AES fault attacks (*e.g.* [11], [9], [7], [3]) that target an entire byte, regardless the number of faulty bits. The experiments reported in this paper also apply to other attacks (*e.g.* [11], [9]) and underline the possibility to *precisely* modify even flags, counters, pointers and other single memory cells that control program flow, in the absence of countermeasures, even at a single-bit level.

In summary, this note's main conclusions are:

- It is possible to implement a single-bit laser fault attack on an AES round key.
- Even when it is physically impossible to target a single-bit on one byte because the beam hits a few other bytes, careful spatial and temporal coordination may allow to deceive the encryption process to consider logically only a single-bit fault that corresponds to Giraud's bit scheme.
- It is possible to reproduce the *same faults* on different plaintexts. This assesses the reality of bit-level Giraud's scenario on unprotected chips.

References

[1] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall and C. Whelan, *The sorcerer's apprentice guide to fault attacks*, Proceedings of the IEEE, vol. 94 (2), IEEE, 2006, pp. 370–382.

[2] E. Biham and A. Shamir, *Differential fault analysis of secret key cryptosystems*, Proceedings of Crypto'97, LNCS, vol. 1294, Springer-Verlag, 1997, pp. 513–525.

[3] J. Blömer and J.P. Seifert, *Fault based cryptanalysis of the Advanced Encryption Standard (AES)*, Financial Cryptography – Proceedings of FC 2003, LNCS, vol. 2742, Springer-Verlag, 2003, pp. 162–181.

[4] M. Brstle *et al.*, SOSSE – *Simple Operating System for Smartcard Education*, www.mbsks.franken.de/sosse/index.html.

[5] G. Canivet, *Analyse des effets d'attaques par fautes et conception sécurisée sur plate-forme reconfigurable*, Ph.D. thesis, Institut polytechnique de Grenoble, 2009.

[6] F. Darracq, T. Beauchêne, V. Pouget, H. Lapuyade, D. Lewis, P. Fouillat and A. Touboul, *Single-event sensitivity of a single SRAM cell*, IEEE Transactions on Nuclear Science, vol. 49 (3), IEEE, 2002, pp. 1486–1490.

[7] P. Dusart, G. Letourneux and O. Vivolo, *Differential fault analysis on* A.E.S., Proceedings of the Int. Conf. on Applied Cryptography and Network Security – ACNS 2003, LNCS, vol. 2846, Springer-Verlag, 2003, pp. 293–306.

[8] Ch. Giraud, DFA *on* AES, Proceedings of AES 2004, LNCS, vol. 3373, Springer-Verlag, 2005, pp. 27–41.

[9] A. Moradi, M.T. Manzuri Shalmani and M. Salmasizadeh, *A generalized method of differential fault attack against* AES *cryptosystem*, Cryptographic Hardware and Embedded Systems – Proceedings of CHES 2006, LNCS, vol. 4249, Springer-Verlag, 2006, pp. 91–100.

[10] National Institute of Standards and Technology (NIST), *Announcing the advanced encryption standard (AES)*, Federal Information Processing Standards Publication, vol. 197, 2001.

[11] G. Piret and J.J. Quisquater, *A differential fault attack technique against SPN structure with application to the* AES *and* KHAZAD, Cryptographic Hardware and Embedded Systems – Proceedings of CHES 2003, LNCS, vol. 2779, Springer-Verlag, 2003, pp. 77–88.

[12] V. Pouget, *Test et analyse par faisceau laser : Plateforme et applications*, Journée thématique du GDR SOC-SIP, 2007. www.lirmm.fr/soc_sip/6fev/GCT_R1_Pouget.pdf

[13] S. P. Skorobogatov and R. J. Anderson, *Optical fault induction attacks*, Cryptographic Hardware and Embedded Systems – Proceedings of CHES 2002, LNCS, vol. 2523, Springer-Verlag, 2002, pp. 2–12.

[14] A. Tria, B. Robisson, J.M. Dutertre and A.P. Mirbaha, *Fault attacks from theory to practise: what is possible to do?*, 2-nd Canada-France Workshop on Foundations & Practice of Security, 2009. www-mitacs2009.imag.fr/Material/mitac_part1.pdf and mitac_part2.pdf