Frontside Laser Fault Injection on Cryptosystems – Application to the AES' last round

Cyril Roscian, Jean-Max Dutertre, Assia Tria

Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on
http://dx.doi.org/10.1109/HST.2013.6581576

Frontside Laser Fault Injection on Cryptosystems – Application to the AES' last round –

Cyril Roscian*,Jean-Max Dutertre* and Assia Tria†
*†Département Systèmes et Architectures Sécurisées (SAS)
*École Nationale Supérieure des Mines de Saint-Étienne (ENSMSE), †CEA-TECH, Gardanne, France
{roscian, dutertre}@emse.fr {assia.tria}@cea.fr

***Abstract*—Laser fault injection through the front side (and consequently the metal-fills) of an IC is often performed with medium or small laser beams for the purpose of injecting bytewise faults. We have investigated in this paper the properties of fault injection with a larger laser beam (in the 100μm range). We have also checked whether the *bit-set* (or *bit-reset*) fault type still holds or whether the *bit-flip* fault type may be encountered. Laser injection experiments were performed during the last round of the Advanced Encryption Standard (AES) algorithm running on an ASIC. The gathered data allowed to investigate the obtained fault models, to conduct two usual Differencial Fault Attack (DFA) schemes and to propose a simple version of a third DFA.**

***Index Terms*—DFA, laser fault injection, fault model, AES**

I. Introduction

Secure circuits are prone to a wide range of physical attacks. Among them, fault attacks (FA) are based on the disturbance of the chip environmental conditions in order to induce faults into its computations. Fault injection may be achieved by using laser exposure [1] [2], voltage [3] or clock glitches [4], electromagnetic perturbation, etc. It exists a very efficient method called Differential Fault Attack (DFA) applied to encryption algorithms that takes advantage of a comparison between correct and faulted ciphertexts to retrieve the secret key used during the ciphering process. These different attack schemes involve strong constraints on the faults location, range, and injection-time. Nevertheless laser injection is often considered as one of the best means to inject faults in order to perform a DFA. Indeed, a laser source allows a precise control on repeatability, timings of injection (the shot instant and pulse duration) and focalization. It appears as a suitable tool to meet the constraints of the various DFA schemes. However, since the technology of Integrated Circuits (IC) is continuously evolving (more transistors are inside the effect area of a given laser beam, and more metal-fills are reflecting it) this statement has to be checked.

In this paper, we studied the effect of a large laser spot to inject faults into the calculations of our target: an ASIC implementing the AES [5] algorithm. We also analysed the effects of front side injection on the properties of the injected faults (fault type, repeatability, fault range, etc.) The obtained data were used to perform two usual DFA [6][7]. Then, we simplified an existing DFA [8] that allowed us to perform the analysis with less complexity. This approach took advantage of the experimental settings (i.e. the use of a large laser spot through the front side and its metal-fills).

The paper is organized as follows. The first part is a reminder of the different effects of laser on silicon: it emphasizes on the notion of laser-sensitive areas, and also gives a description of the fault injection process. The laser set-up and the device used for the test are described in the second part followed by the display of the experimental results and their analysis about the observed fault model and its justification. The third part reports the use of two usual DFA on the experimental results and the simplification of a third DFA to enhance the efficiency. Finally, all these results are summarized in the conclusion with some perspectives.

II. Fault injection with a Laser source

Laser shots on ICs were firstly used to simulate radiation-induced faults [9]. More recently the use of a laser to inject faults into the computations of a secure device was introduced by S. Skorobogatov and R. Anderson [1]. In the following we first remind the main properties of the photoelectric effect created by a laser passing through silicon before describing the resulting fault injection process.

A. Laser effects on ICs and consequences

The photoelectric effect is generated by a laser beam passing through silicon provided that its photons energy is greater than the silicon bandgap [9]. This effect creates electron-hole pairs along the laser path. Generally these pairs recombine and there is no noticeable effect on the IC's behaviour. However, under specific conditions, some undesired effects may appear: the so-called Single Event Effects (SEE).

A SEE happens when the charge carriers (i.e. electrons and holes) created by the laser beam are drifted in opposite directions by the electrical field found in the PN-junctions of CMOS transistors instead of recombining. As a consequence a transient current (i.e. moving charge carriers) is generated through the struck junction. This phenomenon is depicted in the left part of Fig. 1, where the PN-junction of an NMOS transistor in its "turned OFF" state is drawn. After the creation of the electron-hole pairs along the laser beam, two phenomena lead to the creation of the transient current: the prompt charge collection, or funnelling, and the diffusion. The first phenomenon stretches the depletion region (hence the extension of the electric field) along the laser beam, within

a few picoseconds the charges nearby are collected giving a current peak. Then, in a second time, the remaining charges are collected in a longer diffusing scheme: the diffusion. The right part of Fig. 1 shows the transient current associated with these two phenomena as given in [10].

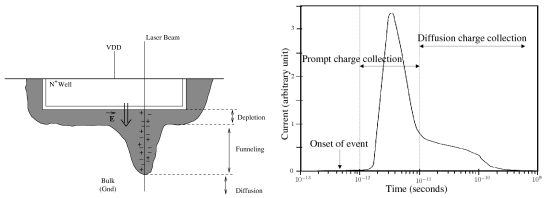

Fig. 1. Photoelectric effect of a laser beam through a PN-junction (left) - Transient current resulting from charge collection after a laser shot [10] (rigth).

It exists a strong electric field, sufficient to create a transient current as explained above, in any PN-junction of the transistors used in CMOS logic regardless of their state (i.e. turned "ON" or "OFF"). However, such a transient current may, or may not, have an effect on the target's logic signals depending on both its location and the data handled by the logic. These dependencies are usually explained by considering the inverter case (see figure 2).

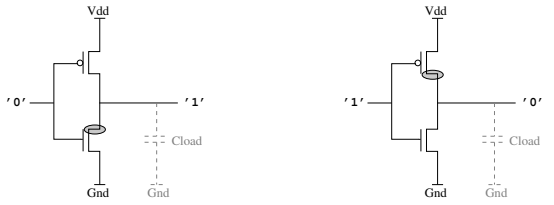

Fig. 2. Inverter's schematic with its data-dependent sensitive areas.

Consider the left part of Fig. 2 where the inverter's input is at a low logical level: its PMOS transistor is turned "ON" and its NMOS transistor is turned "OFF". Hence the inverter's output is at a high logical level and its output's capacitive load (dotted in Fig. 2) is charged. The inverter has four PN-junctions which are likely to give rise to a transient current if struck by a laser: the drains and sources of both PMOS and NMOS transistors. Nevertheless only a transient current originated in the NMOS' drain will result in a disturbance of the inverter's output (pointed out by a filled grey ellipse). In that case, the transient current is flowing from the drain to the substrate which is grounded (as drawn in the top part of Fig. 1). Hence the capacitive load is discharged provided that the transient current is big enough to overcome a charging current flowing through the "ON" PMOS transistor. As a result the output of the inverter passes temporarily to a low logical level. When the transient current vanishes, the capacitive load is charged again via the turned "ON" PMOS transistor. Thus, due to the transient current generated in the NMOS' drain, the output voltage of the inverter undergoes a transient voltage inversion. This transient voltage may then propagate through the downstream logic: a so-called Single Event Transient (SET). Any transient current created in the NMOS' source has no effect on the output since it is isolated from the output by the turned "OFF" NMOS. Regarding the transient currents created in the PMOS' diffusions, they create a leakage path to the N-well which is biased at the core supply voltage (i.e. Vdd). Hence they have no discharging effect on the output's capacitive load. To sum up, the only laser, or SEE, sensitive area of an inverter, when its input is in a low logical state, is the drain of the "OFF" NMOS transistor.

Likewise, when considering an inverter with its input at high level (right part of Fig. 2), a similar reasoning may be conducted. It results that the only laser, or SEE, sensitive area of an inverter when its input is in a high logical state is the drain of the "OFF" PMOS (underlined in grey).

As a conclusion, the laser sensitive area of a CMOS inverter is the drain of the "OFF" transistor, whose location is changing with the logical level of the inverter's input. In a more general way the laser sensitive areas of CMOS ICs are data-dependent. The occurrence of a laser-induced fault depends on the handled data.

B. Laser fault injection mechanism

As reminded in the previous section the laser illumination of an IC's sensitive area results in the propagation of a transient voltage in its logic. This SET may turn into a computational fault according two mechanisms.

The first mechanism is illustrated in figure 3. The SET becomes a fault as it is latched into a register (or D flip-flop, DFF) in place of the correct data. If it reaches the DFF's input outside its latching window (around the rising edge of the clock) the SET vanishes without any effect on DFF's output value and then on the target's calculations (denoted 1^{st} case in Fig. 3). On the contrary if it reaches the DFF's input during the latching window, it is latched: a fault is actually injected (denoted 2^{nd} case in Fig. 3) and change the DFF's output value.

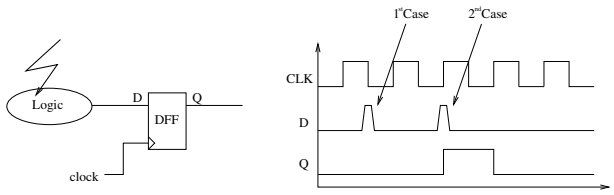

Fig. 3. Fault injection mechanism due to a SET.

The second mechanism happens when the SET is generated inside a DFF or an SRAM. These memory elements are indeed made of cross-coupled inverters. As a consequence the SET will propagate from the struck inverter's output to its own input by passing through the cross-coupled inverter. As a result, the memorised data is inverted: a fault is injected. This fault injection mechanism is called a Single Event Upset (SEU).

Fault injection according these two mechanisms is also time-dependent. This statement is obvious regarding the propagation of an SET as exemplified in Fig. 3. A similar behaviour takes place for SEUs. The resulting false data, which is stored in a DFF, has to propagate and to induce at least one

miscalculation in the logic. Then, it has to be latched into the subsequent registers bank to be turned into an actual fault. This process has timing requirements: if the SEU arises before and to close to the clock rising edge it will be soon overwritten by a correct data. Thus it may not have the time to reach the next register bank before the clock rising edge. Consequently the false data may propagate through the downstream logic nearly followed by a correct data: it may be overwritten before being latched at the next clock rising edge.

C. Discussion on the laser fault model

The properties of laser-induced faults reviewed in subsections II-A and II-B were established under the assumption that the effect range of the considered laser spots was circumscribed to one sensitive area. Under this assumption the fault injection process is data-dependent. More precisely the sensitive areas are changing with the data. For a given laser setting (location of the laser spot, energy level, timing of the injection), the fault may occur or not depending of the data processed by the target. This behaviour may be described as a *bit-set* or a *bit-reset* fault type [11]. A data bit suffers from a *bit-set* (resp. a *bit-reset*) fault, if it is changed from 0 to 1 (resp. from 1 to 0), thus creating a calculation error. On the contrary, it remains unfaulted if its logical value was yet a 1 (resp. a 0). This fault type is very worrying as it makes it possible to mount safe error attacks against cryptosystems [12].

The ability to obtain a *bit-set* or *bit-reset* fault in former technologies is well established [13]. However, this ability is questionable in advanced CMOS processes. The first reason is that the minimal diameter of a laser spot could not be successfully decreased to smaller than 1μm due to optical constraints. Hence, in advanced technologies the laser spot could encompass several transistors violating our first assumption and induce a *bit flip* fault type (which refer to an inversion of the faulted bit regardless of its value) or impact several bits. Moreover as technologies are evolving the metal density over ICs increases due to metal-fill requirements [14]. Metal lines or fills are reflecting laser beams making it more and more difficult to access to sensitive areas from the front side. The main consequence is that most of laser fault injection are carried out through the rear side with a small spot size. This method is not easy, time consuming and a proper preparation of the chip (i.e. de-packaging) is needed. In the other side, laser fault injection, in front side with a large spot, is easier to perform but seems to be not consistent with *bit-set* or *bit-reset* fault injection. This is one point we have considered to explore in this work: the effect of a large laser spot ($\sim 125\mu m * 125\mu m$) through the metal coverage on the fault properties. The *bit-flip* fault type was also considered in this work.

III. Experimental Setup

A. The Laser test bench

The fault injection experiments reported in this paper were performed front side with a green laser source of 532nm wavelength. The laser pulses it produces, have a constant duration of 5ns. Optical settings were chosen to obtain a square laser spot of $125\mu m * 125\mu m$. The energy was tuned to 750nJ per shot. Hence, given the transmission coefficient of the optics, an energy density of 17pJ/μm^2 was achieved under the laser beam. For the sake of simplicity a trigger signal issued from the test chip was used to synchronize the laser shots with the AES encryptions. It made it possible to target the beginning of the AES last round with a jitter of $\stackrel{+}{-}5ns$.

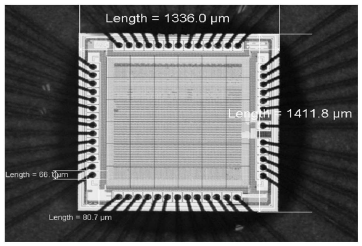

Fig. 4. The AES-128 target device with its 36 shooting sectors

B. The target device

The target device is an ASIC implementing the AES algorithm [5] in its 128-bits key length version (AES-128). The AES-128 is a substitution and permutation algorithm. It consists of 10 identical rounds after a short initial round; each round is a succession of four different transformation. In our implementation, one round needs only one clock cycle to be achieved. The entire encryption is performed in 11 clock cycles. The ASIC was operated at 25 MHz during our experiments (although its maximum allowable frequency is 50 MHz). A picture of the silicon chip is given in figure 4. It was designed in a 6-metal layers, 0.13 μm technology. None of the circuit's functional blocks is identifiable at sight: the whole design was scrambled (glue logic). Fig. 4 also shows the partitioning of the target's surface into 36 shooting sectors corresponding of the laser spot's size. Laser fault injections were done according this partitioning.

C. Experimental results

In the following, the obtained faults are classified and analysed at byte level because they were induced during the AES' last round, corresponding cryptanalyses of which are considered byte-wise (see section IV).

A large fault injection experiment was conducted. For every shooting sector of Fig. 4, 10,000 encryptions were ran with random plaintexts but the same key. During these experiments no latch-up or reset of the component have been reported. Simultaneously the laser was fired and the output ciphertext (faulted or not) was retrieved. It was then analysed to establish whether it was erroneous or not. In case of an error the injected

fault was recovered by reversing the encryption of the faulted ciphertext and also by comparison with a correct encryption (the key and the plaintexts were known). Despite the large spot size ($125\mu m * 125\mu m$), most of the induced faults were single-bit faults. Table I reports a synthesis of these results. For every byte of the AES state the faults of the shooting sector, corresponding to the highest error rate, are reported. The rates of the single-bit and most common faults are also given (e.g. consider byte$_0$: 480 encryptions were faulted, amongst them 79% were single-bit faults and the most encountered fault appeared with a 74% rate).

TABLE I
EXPERIMENTAL RESULTS

byte #	Error injection rate	Single-bit error rate	Most common fault rate
0	4.8%	79%	74%
1	3.2%	100%	99%
2	3.1%	98%	92%
3	67.8%	49%	48%
4	9.4%	99.7%	90%
5	2.1%	79%	58%
6	0.5%	100%	99%
7	4.6%	65%	64%
8	23%	64%	42%
9	7.2%	91%	80%
10	4.3%	99%	98%
11	15.,5%	97%	97%
12	12.2%	98%	96%
13	3.1%	87%	55%
14	0.2%	100%	100%
15	7%	99.2%	99%

A second set of experiments was conducted for a location of the laser beam that gave rise to a relatively high error occurrence rate on byte number 5. 1,000 encryptions with the same previous constant key and random plaintexts were done simultaneously with laser injection. Its results are reported in table II.

TABLE II
LASER INJECTION ON BYTE NUMBER 5 (RANDOM PLAINTEXTS)

Faults occurrence rate	Occurrence rate of fault '0x80'	Occurrence rate of other faults
7.1%	94%	6%

One of the previous plaintexts related to an actual injection of fault was then selected to conduct another set of 1,000 encryptions (i.e. the whole experimental settings were the same for each attempt). Table III displays its statistics.

TABLE III
STATISTICS OF FAULT INJECTION ON A CONSTANT PLAINTEXT

Faults occurrence rate	Occurrence rate of fault '0x80'	Occurrence rate of other faults
16.8%	97%	3%

D. Analysis of the laser-induced faults

The fault occurrence rate reported in table III (for one given plaintext) is approximately twice the rate reported in table II (for several random plaintexts). The explanation lies in the data-dependent nature of laser-induced faults. A deeper analysis of the single-bit fault (i.e. 0x80) injected during these experiments revealed that the faulted bit was always a 0 turned into a 1. Moreover,among the 1,000 encryptions of table II no fault was injected when the original value of the faulted bit (i.e. the correct, or none faulted, bit value) was a 1. These results reveal a *bit-set* fault type. Moreover the faults occurrence rate grows from 7.1% to 14.2% when only the fault injection attempts consistent with a *bit-set* are considered.

However, a 16% faults occurrence rate is still low as laser fault injection is often considered as deterministic. The jitter of the laser setup is probably an (incomplete) explanation given the time-dependent nature of the fault injection mechanism (see subsection II-B). The fact that several laser-sensitive areas are under the laser spot is another hypothesis that would be worth studying.

Among the large amount of data we have processed, we also report here a further analysis of the result obtained on byte #3 during the first experiment. They are extensively reported in table IV. At bit-level, the fault occurrence rates of bits b_2 and b_1 were respectively 34.3% and 66%. A deeper analysis revealed that the fault types of bits b_2 and b_1 were respectively a *bit-set* and a *bit-flip*.

TABLE IV
FAULTS INJECTED ON BYTE NUMBER 3

Fault value $b_7...b_4$ $b_3b_2b_1b_0$	# of occurrence
0000 0110	3285
0000 0010	3228
0000 1110	93
0000 1000	70
0000 0100	51
0000 0001	40
0000 1001	13
0000 0011	4

These experiments showed that the *bit-set* (or *bit-reset*) and *bit-flip* fault types are both attainable for laser fault injection with a large spot from the front side of the target. Our explanation is that the metal-fills act as *shutters* that allow to stimulate only a few laser-sensitive areas. In some cases it is consistent with the *bit-set* or *bit-reset* fault type. For others two sensitive areas related to a common data bit are simultaneously stimulated: one by a *bit-set* fault type, the second by a *bit-reset*. As a result a *bit-flip* fault type is achieved. An intended application of this phenomenon is reported in [14].

IV. DFA OF THE AES LAST ROUND

Two schemes of DFA are based on an analysis of the AES last round in the presence of faults [6], [7]: they make it possible to retrieve its last round key. In this section we study the relevance of using these techniques to process our experimental data, especially with the data where the single-bit fault occurrence rate or repeatability are low. We also report and discuss the use of a DFA scheme on the AES last round recently introduced by [8] with these data.

A. Notations

In the following M, C, D, K and E denote respectively the plaintext, the correct ciphertext, the faulty ciphertext, the secret key and the error value (calculated by Xoring C and D). Depending on the context they may refer to the whole AES state (16 bytes) or to a given single byte (corresponding to the fault location). The *SubByte* transformation of the AES is denoted SB. The AES state at the beginning of the j^{th} round will be denoted by $Mj-1$ (e.g. $M9$ denotes the state at the start of the last round). Kj refers to the round key of the j^{th} round. For the sake of clarity the *ShiftRows* transformation is left out in the following equations. A subscript index i may be used to point to a given encryption among others. As there is no *MixColumns* transformation during the AES last round the cryptanalyses are performed bytewise.

B. Application of the Giraud's DFA

Giraud [6] has introduced the first DFA against the AES last round. It is based on a single-bit fault model, whose faults have to target $M9$. The attack is performed bytewise: a byte of $K10$ is retrieved with a success rate of 97% from three pairs of correct and faulty ciphertexts (C_i, D_i) by solving equations 1 and 2.

$$C_i = K10 \oplus SB(M9_i) \tag{1}$$

$$C_i \oplus D_i = SB(M9_i) \oplus SB(M9_i \oplus E) \tag{2}$$

With the experimental results presented in III-C, only three byte of the AES state are compliant with the single-bit fault model of Giraud's DFA (bytes 1, 6, and 14 in tab. I). 13 bytes are close or beyond a single-bit occurrence rate of 80%. Hence a higher number of (C_i, D_i) pairs is needed to find the corresponding key bytes. However, the attack is still feasible.

Yet two bytes have single-bit rates around 65% and byte #3 is below 50%. For bytes presenting these statistics, Giraud's DFA may not be the most efficient scheme.

C. Application of the Roche et al. DFA

The DFA recently introduced by Roche et al. [7] was originally based on the injection of constant faults on the 9^{th} and 10^{th} round keys. From the equations of a correct and a faulty ciphertexts (C_i and D_i respectively) equation 3 is obtained:

$$SB(SB^{-1}(C_i \oplus K10) \oplus E9) \oplus K10 \oplus E10 = D_i \tag{3}$$

where $E9$, $E10$, and $K10$ are unknown. The cryptanalysis is conducted bytewise. The success rate in retrieving $K10$ is higher than 90% with three pairs (C_i, D_i).

This technique was also extended to none-constant faults: as the fault repeatability is decreasing, the number of (C_i, D_i) pairs increases. Moreover, without any lack of generality, this attack may be expanded to the fault model used in our experiments (i.e. faults injected on $M9$) by nullifying $E10$ in eq. 3.

The experimental data gathered in our experiments were analysed according this DFA scheme. Given the fault repeatability (i.e. the most common fault rate of table I's last column) 9 bytes required 6 or less (C_i, D_i) pairs to retrieve the corresponding key bytes because their fault repeatability was higher than 90%. Four bytes revealed a fault repeatability around 50%: they required 15 (C_i, D_i) pairs on average to discover their round key bytes.

Likewise Giraud's DFA, this scheme makes it possible to retrieve the secret key. Despite this required more data to succeed. However, the corresponding fault model is less constraint than the single-bit requirement of Giraud's DFA. It may succeed where Giraud's DFA will fail.

D. Simplification of an existing DFA

Lashermes et al. [8] have introduced a DFA scheme that makes use of faults injected on $M9$ at the beginning of the AES last round. Its originality compared with the DFA schemes of Giraud and Roche et al. resides in the bytewise analysis of the injected faults. From the equations of the correct ciphertext C_i and the faulted ciphertext D_i (equations 4 and 5 respectively):

$$C_i = K10 \oplus SB(M9_i) \tag{4}$$

$$D_i = K10 \oplus SB(M9_i \oplus E_i) \tag{5}$$

where E_i is the injected fault, the expression of E_i (eq. 6) is obtained:

$$E_i = SB^{-1}(C_i \oplus K10) \oplus SB^{-1}(D_i \oplus K10) \tag{6}$$

In eq. 6 C_i and D_i are known and E_i and $K10$ are unknown. The DFA consists in building an error table (see table V): its columns represent the 256 round key feasible values (i.e. hypotheses k); for each line (called a realization of index i) the corresponding values of the injected fault $e_{i,k} = E_i$ are calculated from eq. 6, the (C_i,D_i) pairs, and the key hypothesis k.

TABLE V
ERROR TABLE

	K10 hypothesis k				
Realization i	'0x00'	'0x01	'0x02'	$\cdots$	'oxFF'
0	$e_{0,0}$	$e_{0,1}$	$e_{0,2}$	$\cdots$	$e_{0,255}$
1	$e_{1,0}$	$e_{1,1}$	$e_{1,2}$	$\cdots$	$e_{1,255}$
2	$e_{2,0}$	$e_{2,1}$	$e_{2,2}$	$\cdots$	$e_{2,255}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Only one column of the error table corresponds to the correct key byte $K10$. This column also gives the faults that have been actually injected. [8] provides a complete methodology based on the calculation of the entropy of the errors to discriminate the right key byte (i.e. the right column).

Table VI reports the error table obtained from the fault injection attempts on byte #3 (cf. tab. IV).

It is therefore easy to ascertain that the corresponding value of the key byte is '0xCD'. The faults values in every columns appear random except for the key hypothesis '0xCD' where

TABLE VI
ERROR TABLE OF BYTE #3

	K10 hypothesis k					
Realization i	'0x00'	'0x01'	···	**'0xCD'**	···	'0xFF'
0	'0x63'	'0x61'	···	**'0x02'**	···	'0x15'
1	'0xB2'	'0x0A'	···	**'0x06'**	···	'0x59'
2	'0x0C'	'0xBF'	···	**'0x02'**	···	'0x1E'
···	···	···	···	···	···	···
158	'0x51'	'0xFF'	···	**'0x06'**	···	'0x1A'
···	···	···	···	···	···	···
3,578	'0xF2'	'0x49'	···	**'0x08'**	···	'0x82'
···	···	···	···	···	···	···
10,000	'0x09'	'0x3B'	···	**'0x0A'**	···	'0x33'

the faults (outlined in bold) are restricted to the four least significant bits (see Table IV). This result is consistent with the design of our target IC: only four bits of byte #3 were affected by the laser beam because its logic blocks are scrambled.

Only a few (C_i,D_i) pairs may be sufficient to find $K10$'s byte. Figure 5 excerpted from [8] gives the average number of faults needed to succeed as a function of the faults entropy.

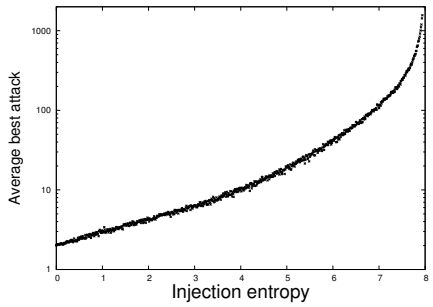

Fig. 5. Average minimum number of faults needed to find the key for a given injection entropy [8]

The fault injection process on byte #3 has an entropy of 1.3. Thus 3.5 faults on average are needed to successfully retrieving the right key byte. Given the statistics of byte #3 ($\sim 50\%$ repeatability and single-bit occurrence rate) this approach appears more efficient than Giraud's and Roche et al.'s DFA (15 realizations are needed for the latter). However, this statement cannot be generalised. It holds when the injected fauls have a distinctive pattern and a low repeatability.

V. CONCLUSION AND DISCUSSION

We have described in this paper experiments of laser fault injection through the front side of an IC implementing the AES-128. Because of the reflective effect of its metal fills a large laser beam ($125 * 125\mu m^2$) was used. Injection with a laser beam of a few micrometers would have been time-overconsuming.

We have observed two fault types: *bit-flip* and *bit-set* (or *bit-reset*). The latter type was unexpected because it seemed more consistent with the use of a smaller beam affecting only one laser-sensitive area. The *bit-flip* type is explainable by the simultaneous illumination of two sensitive areas corresponding respectively to a *bit-set* and a *bit-reset*. These results are obtained because the metal fills behave as *shutters*: at byte level only one or very few sensitive areas are exposed to the laser. Moreover a large part of the induced faults were single-bit despite the size of the laser beam. This precision was achieved thanks to the metal coverage. The analysis of the injected faults had also corroborated the data-dependent and time-dependent nature of laser injection.

These fault injection experiments were performed at the beginning of the AES last round. The faults statistics allowed to recover the secret key by using either Giraud's DFA [6] or Roche et al. DFA [7]. However, for faults with a low single-bit occurrence rate and/or a low repeatability, these two schemes of DFA are not the most efficient. We finally proposed a simple application of the DFA scheme introduced by Lashermes et al. [8] that makes it possible to recover the secret key in a more efficient way.

ACKNOWLEDGEMENT

The research work of Cyril Roscian was partly funded by the "Conseil Regional PACA". The authors also would like to thank Ronan Lashermes for his help and his support.

REFERENCES

[1] Skorobogatov, S., Anderson, R.: Optical fault induction attacks. In: Cryptographic Hardware and Embedded Systems - CHES 2002. Volume 2523 of Lecture Notes in Computer Science. (2003) 31–48
[2] van Woudenberg, J., Witteman, M., Menarini, F.: Practical optical fault injection on secure microcontrollers. In: Fault Diagnosis and Tolerance in Cryptography (FDTC), 2011 Workshop on. (2011) 91–99
[3] Blömer, J., Seifert, J.P.: Fault based cryptanalysis of the advanced encryption standard (aes). In: Computer Aided Verification. Volume 2742 of Lecture Notes in Computer Science. (2003) 162–181
[4] Agoyan, M., Dutertre, J., Naccache, D., Robisson, B., Tria, A.: When clocks fail: On critical paths and clock faults. Smart Card Research and Advanced Application (2010) 182–193
[5] NIST: Announcing the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication, n. 197 (2001)
[6] Giraud, C.: Dfa on aes. In: Advanced Encryption Standard – AES. Volume 3373 of Lecture Notes in Computer Science. (2005) 571–571
[7] Roche, T., Lomné, V., Khalfallah, K.: Combined fault and side-channel attack on protected implementations of aes. In: Smart Card Research and Advanced Applications. (2011) 65–83
[8] Lashermes, R., Reymond, G., Dutertre, J.M., Fournier, J., Robisson, B., Tria, A.: A dfa on aes based on the entropy of error distributions. In: 9th Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC). (2012)
[9] Habing, D.H.: The use of lasers to simulate radiation-induced transients in semiconductor devices and circuits. In: Nuclear Science, IEEE Transactions on. Volume 12. (1965) 91 –100
[10] Wang, F., Agrawal, V.: Single event upset: An embedded tutorial. In: Proc. of 21st International Conference on VLSI Design. (2008) pp. 429–434
[11] Otto, D.: Fault Attacks and Countermeasures. PhD thesis, Paderborn University (Germany) (2004)
[12] Loubet-Moundi, P., Vigilant, D., Olivier, F.: Static fault attacks on hardware des registers. Cryptology ePrint Archive, Report 2011/531 (2011)
[13] Pouget, V., Fouillat, P., Lewis, D., Lapuyade, H., Sarger, L., Roche, F., Duzellier, S., Ecoffet, R.: An overview of the applications of a pulsed laser system for seu testing. In: On-Line Testing Workshop, 2000. Proceedings. 6th IEEE International. (2000) 52 –57
[14] Balasubramanian, A., McMorrow, D., Nation, S., Bhuva, B., Reed, R., Massengill, L., Loveless, T., Amusan, O., Black, J., Melinger, J., Baze, M., Ferlet-Cavrois, V., Gaillardin, M., Schwank, J.: Pulsed laser single-event effects in highly scaled cmos technologies in the presence of dense metal coverage. Nuclear Science, IEEE Transactions on **55** (2008) 3401 –3406