Electromagnetic Transient Faults Injection on a hardware and software implementations of AES

Amine Dehbaoui, Jean-Max Dutertre, Bruno Robisson, Assia Tria

Fault Diagnosis and Tolerance in Cryptography (FDTC), 2012 Workshop on
DOI: 10.1109/FDTC.2012.15

Electromagnetic Transient Faults Injection on a hardware and software implementations of AES

Amine Dehbaoui*, Jean-Max Dutertre†, Bruno Robisson* and Assia Tria*

*†*Département Systèmes et Architectures Sécurisées (*SAS*)*
†École Nationale Supérieure des Mines de Saint-Étienne
*CEA-LETI, *Gardanne, France*
{Firstname.Lastname}@cea.fr Lastname@emse.fr

***Abstract*—This paper considers the use of electromagnetic pulses (EMP) to inject transient faults into the calculations of a hardware and a software AES. A pulse generator and a $500\mu m$-diameter magnetic coil were used to inject the localized EMP disturbances without any physical contact with the target. EMP injections were performed against a software AES running on a CPU, and a hardware AES (with and without countermeasure) embedded in a FPGA. The purpose of this work was twofold: (a) reporting actual faults injection induced by EMPs in our targets and describing their main properties; (b) explaining the coupling mechanism between the antenna used to produce the EMP and the targeted circuit, which causes the faults. The obtained results revealed a localized effect of the EMP since the injected faults were found dependent on the spatial position of the antenna on top of the circuit's surface. The assumption that EMP faults are related to the violation of the target's timing constraints was also studied and ascertained thanks to the use of a countermeasure based on monitoring such timing violations.**

***Keywords*-Electromagnetic Fault, Electromagnetic Pulse, AES, FPGA, MCU.**

I. Introduction

Electronic devices that implement cryptographic features (such as "smart cards") are key components to our information society: they provide secure communications. As a consequence, they are subjects to physical 'attacks'. Among them, *fault attacks* are considered being very powerful. They consist firstly in modifying the behavior of the chip with dedicated experimental setups and secondly in recovering the secret information by using cryptanalysis techniques based on Differential Fault Analysis (DFA) [7], [19], [13], [23], safe-errors [29], fault sensibility analysis [17], collisions [8], round reductions [11], etc.

Various experimental setups are commonly used to modify the behavior of a chip (i.e. fault its computations). Underpowering a device, overheating or over-clocking it [26], [5], [15] lead to set up time violations resulting in the injection of errors. Another means of fault injection is the use of optical radiations: intense white light (e.g. from a flash bulb) or a laser beam [27]. The latter is widely used while assessing the security of cryptographic systems against fault attacks [28] for certification purposes and may offer the ability to inject fault affecting a byte or even a single bit of the sensitive data [1]. Finally, the electromagnetic (EM) channel can also be used to induce faults in digital devices. This channel is already used to conduct passive attacks in order to retrieve sensitive data handled by a secure device (e.g. secret or private key used by a cryptographic algorithm). These observation attacks, based on the eavesdropping of the target's EM emissions, have been investigated by numerous research groups and have given rise to various publications [21], [12], [2].

In this work, we describe the use of the EM channel to carry out active attacks against a software AES running on a CPU and a hardware AES embedded in a FPGA. Compared with most of previous works on electromagnetic faults targeting RSA (see section II), AES is shorter in time and requires more precise faults.
The transient electromagnetic pulses (EMPs) were injected on top of the surface of both targets. By doing so, we intended to:

- report actual fault injections on two typical targets,
- explain the behavior of the faults induced by a very short EM pulse,
- analyse whether the effect of the EMP on the target is global or local,
- find out the mechanism involved in the injection of a fault induced by an EMP.

This article is organized as follows. A short review of the state-of-the-art of EM active attacks is given in section II. We describe the electromagnetic injection bench used to generate EMPs in section III. In section IV we study the effect of a localized EMP injected on top of the surface of a micro-controller while executing the Advanced Encryption Standard (AES). In sections V and VI we study the effect of a localized EMP injected on top of the surface of an FPGA. The goal is to validate the assumption that an EMP induces a timing violation during calculations. As a conclusion, section VII summarizes our findings and draws some prospects.

II. Electromagnetic Faults : State-of-the-Art

The EM medium may be used to conduct active attacks. Two kinds of near-field EM perturbations are usually considered: transient pulses and harmonic emissions.

Concerning EM harmonic emissions, Alaeldine et al. studied the *electromagnetic compatibility* (EMC) of integrated circuits (IC) to near-field injection with frequencies up to 1 GHz [3]. They investigated the effects of both electric and magnetic fields along the x, y, and z axes. Their test circuits were found sensitive to both magnetic and, to a greater extent, electric fields.

Recently, Poucheret et al. [20] considered the effect of a 1 GHz electric field applied to an IC with an embedded ring oscillator (RO). The main component of that electric field was the transverse one (i.e. parallel to the surface of the chip). The perturbation impacted the output frequency of the RO. Monitoring the effect of that perturbation enabled them to draw a cartography of the sensitive areas of the chip. A cross examination between the layout of the device and the cartography demonstrates that the coupling between the injection probe and the circuit lies mainly in the power-ground network (PGN). More recently, an extension of that work in [6] shows that it is possible to lock RO based true random pattern generators (TRNG) on the harmonic injected signal, and thus to control the bias of the TRNG output.

Regarding transient EM pulses, to our best knowledge, two articles report actual results of successful fault injection. Quisquater et al. [22] in 2002 described the use of an active probe to apply an intense and transient magnetic field on a microprocessor. These results in faulting RAM and EEPROM memory cells. Faults on the device's address bus were also obtained. The authors claimed that the fault injection's mechanism involves the creation of an eddy current in the chip. However, they did not provide any evidence of that statement. More recently, in 2007, Schmidt et al. reported the use of a spark generator to fault a CRT-based RSA algorithm running on an 8-bits micro-controller [25]. The injected fault leads to a successful attack as it allows them to factorize the RSA modulus. Besides, the experimental setups of [22] and [25] are characterized by a very large jitter because of the use of a camera flash-gun or of a spark-generator.

III. Experimental setup

In this section, the electromagnetic injection bench used to generate transient EM Pulses is described.

The fault injection bench (Figure 1) is built of: a control PC, the targeted device, a motorized stage, a pulse generator, and a magnetic probe. The target is fixed on the x-y-z motorized stage. Every element of the bench is controlled by the control PC, and the communication with the target is established through a serial port or a smart card reader.

The pulse generator is used to deliver voltage pulses (with low jitter $< 50ps$) to the magnetic coil. It has constant rising and falling transition times of 5ns. The amplitude

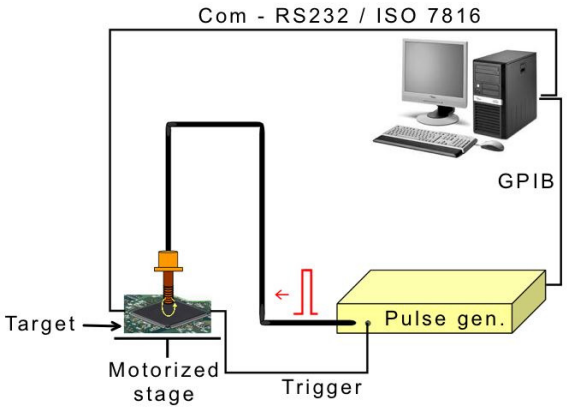

Figure 1. EM pulse injection bench.

range (respectively the width) of the pulses extends from 1V to 100V (respectively from 10ns to 100ns). We used a magnetic probe of diameter $500\mu m$ (Figure 2) in order to disturb only a small part of the targeted device. Note that all the experimental results reported in this paper were obtained identically on open and on untampered packages.

IV. Electromagnetic transient faults on a software implementation of the AES

In this section we study the effect of a localized EMP injected on top of a micro-controller executing a software implementation of the Advanced Encryption Standard (AES).

A. Target description

We used a smart card emulation board composed of an 8-bits AVR Atmega 128 micro-controller implemented in 0.35μm technology with integrated $128KB$ Flash program memory, 4KB EEPROM and 4KB SRAM. This micro-controller has an operating voltage of $4.5-5.5V$ and runs at

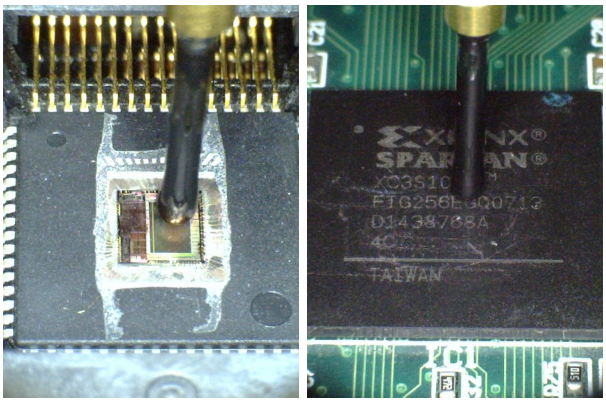

Figure 2. Magnetic injection probe over a micro-controller (left) and a FPGA (right).

a frequency $3.57MHz$. The Atmega has a Harvard architecture, its program and data memories are physically separated. The CPU can load instructions only from Flash program memory and can only write in SRAM. Furthermore, the instruction fetch unit can only access the program memory. As a result, the data memory cannot be executed.
A smart-card-like OS (called SOSSE [9]) is used for communication purposes and a software implementation of the AES encryption algorithm [18] using a key size of 128 bits is embedded. This AES implementation combines the SUBBYTES and the MIXCOLUMNS in a single operation and uses only 8-bits operations on a 4x4 column matrix of bytes termed AES state. Substitution boxes are mapped in memory and all AES round keys are pre-calculated before its execution.

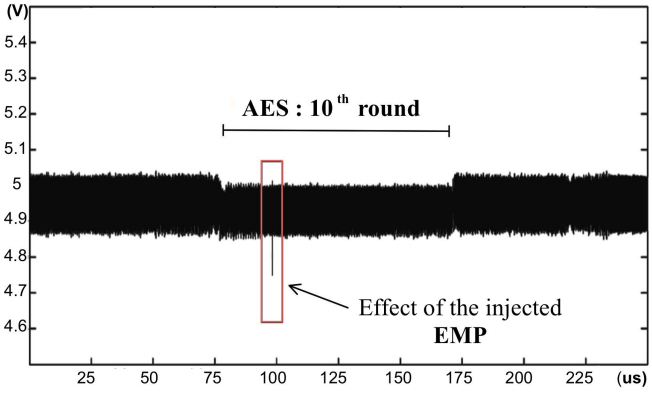

Figure 3. Power supply trace during magnetic fault injection.

B. Injection of transient EM faults

In order to easily identify the faulted bytes in the AES state, we targeted the execution of the last round (i.e. the 10^{th} round for the AES-128). The injection probe was positioned on top of the CPU. In this experiment, a trigger signal labelled "synchronization signal" was generated at the beginning of the round for easy synchronization. Figure 3 shows the power supply of the target during the magnetic fault injection. As we can observe, for an injected EMP of duration $50ns$ and amplitude $50V$, we have obtained a negative spike of less than $50ns$ in width and $150mV$ in amplitude. These voltage variations may seem quite small to be able to induce faults into the device computations. However, because the measurement of the power supply was done on the chip pads, a large part of the perturbation may have been filtered out.
The pulse's width was chosen smaller than the clock period (Tclk = $280ns$) in order to target every instruction executed by the CPU. These parameters are reported in figure 4.

Using the "synchronization signal", we scanned trough the instant of the EMP injection from the beginning to the end of the 10^{th} round (whose duration is 90μs) by steps of $100ns$.

Z position	EMP amplitude	EMP width	clk period	rise/fall times
$< 500\mu$m	$100V$	$50ns$	$280ns$	$5ns$

Figure 4. EMP parameters

At each of these steps, 1,000 encryptions (with the FIPS key) were carried out with and without EMP injection. The results of these two computations were compared and the faulted byte (if any) determined. Depending on the injection time of the EMP, we observed two distinct behaviors regarding the fault value: data-dependent faults (i.e. the injected fault is changed with the plaintext) and constant faults (i.e. the fault value is held constant irrespective of the plaintext). Figure 5 reports for every byte of the AES state, the time at which the EMP was injected and the data-dependency behavior of the faults. It also reports the value (in case of constant fault) and the reproducibility rate of the induced faults.

Faulted byte #	Injection time	Reproducibility	Fault value
0	$0.3\mu s$	100%	Data dependent
1	$9.78\mu s$	100%	Data dependent
2	$19.3\mu s$	100%	Data dependent
3	$33.7\mu s$	100%	Data dependent
4	$55.7\mu s$	100%	Data dependent
5	$12.4\mu s$	100%	Data dependent
6	$63.4\mu s$	100%	Constant "0xFB"
7	$65.9\mu s$	100%	Constant "0x89"
8	$5.53\mu s$	100%	Data dependent
9	$69.5\mu s$	100%	Data dependent
10	$74.5\mu s$	100%	Constant "0x00"
11	$75\mu s$	100%	Data dependent
12	$6.53\mu s$	100%	Data dependent
13	$8.78\mu s$	100%	Constant "0x28"
14	$25.5\mu s$	100%	Data dependent
15	$87.5\mu s$	100%	Constant "0xA6"

Figure 5. Fault values on AES state

As we can observe, the fault injection technique based on the generation of an EMP close to a circuit enables to fault every byte of the AES state. However, the occurrence rate of the induced faults in this experiment depends on the amplitude of the EMP. Figure 6 shows the occurrence rate as a function of the amplitude of the EMP. From the results reported in this figure, the success rate grows with the amplitude of the EMP. We were able to obtain a fault occurrence with a success rate of 100% for a pulse amplitude around $100V$. This latter value depends on the coupling characteristics between the antenna and the metal-top geometry of the chip.

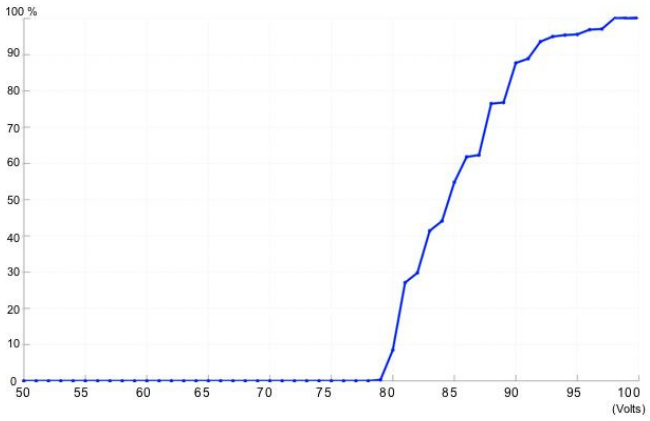

Figure 6. Fault occurrence rate versus amplitude of the voltage pulse

C. Faults analysis

This section deals with the analyze of the effects of EMPs on the instructions being executed by the CPU. Previous works on clock and power glitches applied to micro-controllers were published. Choukri et al. [11] used power glitches to reduce the number of rounds of an AES implementation. Kim et al. [16] also used power glitches to skip subroutine calls in a software RSA-CRT implementation. Similarly, Schmidt et al. [24] prevented a subroutine call in a square-and-multiply RSA software implementation. More recently Balasch et. al [4] performed a study of the clock glitch effects on the same AVR micro-controller used in our experiment. They showed that instructions can be replaced or skipped by injecting a clock glitch, and that the effects of faults are deterministic and reproducible. More precisely, as the clock period decreases, a larger number of bits of the opcode are stuck at zero.

Considering the last round of the AES, the ADDROUNDKEY, SUBBYTES and SHIFTROWS operations are executed one after the other. The code fragments the figures 7 and 8 are provided in order to explain the fault injection mechanism. The output register containing the final state is reseted at each encryption.

```
1 LDD   R24, Y+i      // load subkey
2 LD    R25, X        // load state
3 EOR   R24, R25      // ExclusiveOR
4 STD   Z+i, R24      // store result
```

Figure 7. AddRoundKey opcodes

For example, if we consider the 10^{th} byte of the AES state (see fig. 5), the induced fault at $74.5\mu s$ is constant (i.e data independent) and equal to 0X00. According to the time of the EMP injection, the operation involved in this case is the ADDROUNDKEY. If we look to the associated code fragment (Figure 7), we can observe that the value 0X00 of the fault corresponds to the value of the output register (0X00 at the initialization). An explanation is that the store instruction (line 4 of fig. 7) has not been executed or has been replaced by a NOP: the same behavior as described in [4].

A particularly interesting case is observed when considering bytes 7 and 15. The induced faults are also constant and respectively equal to 0X89 and 0XA6. Considering the EMP injection time, the operation involved is the ADDROUNDKEY. We can observe that the values of the induced fault are respectively equal to the sub-keys of the FIPS AES key. In this case, an explanation is that the EXCLUSIVE OR (line 3 of fig. 7) has not been executed and that the value of the sub-key is not updated by the result of the EOR in the internal register (R24). This non updated value is then stored in the output register (address Z+I). In this case, a simple fault attack consists in targeting this operation for every byte of the AES in order to obtain every sub-key of the AES key.

Now, if we consider bytes 1, 2, 3, 5, 8, 12 and 14, the induced faults are data-dependent (i.e. a different fault is induced for each different plaintext). Considering the EMP injection time, the operation involved in this case is the SHIFTROWS (the corresponding opcodes are given in figure 8). In this figure, X is the address pointed by (R26-R27), Z the address pointed by (R30-R31), Y+I denotes the address of the byte before the SHIFTROWS, Y+K the address of the byte after the SHIFTROWS. Using the faulty cipher and the FIPS key, we were able to run backward the AES encryption step by step from the erroneous ciphertext to retrieve the behavior of the fault. In that case and according to the different values of the induced faults, the explanation is that the load instruction (line 10 of fig. 8) was not executed during the EMP injection. In fact, the fault value corresponds here to the previously stored value in the register R24 (line 5 of fig. 8) which corresponds to the value of a previous byte of the AES state. This latter value is then used to calculate the final state which explains the data-dependency behavior of the fault.

```
1  LDD   R26, Y+i      // load state i address
2  LDI   R27, 0x00
3  SUBI  R26, 0x00
4  SBCI  R27, 0xF5
5  LD    R24, X        // load the state i
6  STD   Y+k, R24      // store the state i
7  LDI   R31, 0x00
8  SUBI  R30, 0x00
9  SBCI  R31, 0xF5
10 LD    R24, Z        // load the state i+1
11 STD   Y+i, R24      // store the state i+1
```

Figure 8. SubBytes and ShiftRows opcodes

D. Summary

The results obtained by the experiments described in this section show that an EMP injected on top of the micro-controller induces faults into CPU calculations. This effect can be deterministic depending on the amplitude of EMP (and reproducible since we are able to obtain a behavior with an occurrence rate of 100%).
Moreover, and after studying the behavior of the induced faults, it seems that the EMP injection prevents the CPU from executing some instructions by affecting the program flow. As reported on the same target by [4], clock glitches (i.e. decrease of one clock period until a fault is induced) has the same instruction skipping effect. This effect was obtained by violation of the target's timing constraints. This is a first sign, however insufficient, to conclude that EMP induced faults may be related to timing constraints violation. The fault occurrence rate depicted in figure 6 is also a feature of faults induced by timing constraints violation that can be split into three distinct behaviors: (a) at low stress (below 78V) no fault is injected; (b) when the stress is progressively increased faults start to be injected (between 79V and 97V) in a non determinist way because of the violation of the register's setup time; (c) then, at high stress (above 97V) the fault injection process is purely determinist. In the latter case, data are latched even before they could have changed.

V. Transient Electromagnetic faults on a hardware implementation of AES

In this section the effect of a localized EMP injected on top of an FPGA is studied. The goal is to validate the assumption that an EMP induces a timing violation during calculations.

A. Target description

The test chip implements a hardware 128 bits version of the AES encryption algorithm. The design was written in VHDL and synthesized for a FPGA from the Xilinx Spartan 3 family. It is built out of three main blocks: a communication and control module, a key expansion module and a cipher module.
The communication and control module is dedicated to the management of a serial link, which receives the plaintext and the key used for the encryption and to transmit the ciphertext. It has also a control task, which consists in driving the two other modules in order to execute correctly the encryption.
We chose to use a 128 bit-wide data path and to execute simultaneously on the chip the key expansion and cipher routines. As a consequence, a complete encryption round takes only one clock period, and the whole encryption process is executed in eleven clock periods.

The key expansion routine generates the round keys "on the fly". For each clock cycle, a new round key is obtained from the key expansion module and sent to the cipher

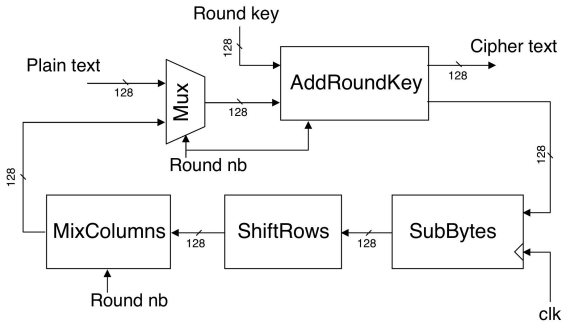

Figure 9. Structure of the cipher module

Z position	EMP amplitude	EMP width	clk period	rise/fall times
$< 500\mu$m	$100V$	$10ns$	$10ns$	$5ns$

Figure 10. EMP parameters

module. The cipher module's architecture is depicted on Figure 9. It is divided into five submodules: AddRoundKey, SubBytes, ShiftRows, MixColumns, and Mux. The first four, as their names suggest, correspond to the individual AES transformations. Note that, as mentioned before, the data path is 128 bit-wide. The AddRoundKey module owns a dedicated output to store the ciphertext after the final round. The MixColumns module is bypassed during the final round.
This module's architecture, shaped in loop, gives a long data propagation path. Consequently, the chip critical delay path is located in the cipher module. The nominal clock frequency of this hardware AES is 100 MHz.

B. Injection of transient EM faults

In this first experiment, the relative distance between the antenna and the surface of the circuit was fixed ($< 500\mu$m), the pulse width was chosen to match the clock period (Tclk = $10ns$). The parameters of that experiment are reported in figure 10. A faults cartography of the design was performed

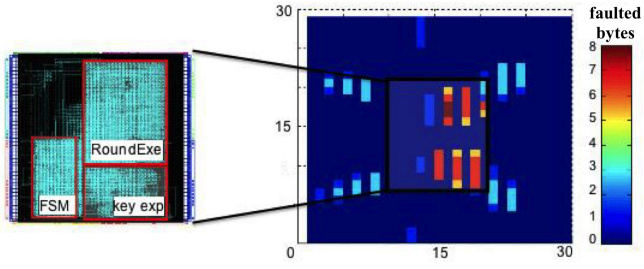

Figure 11. Floorplan of the circuit and associated faults cartography

during the last round of the AES. It was aimed at disclosing the (X,Y) coordinates, for which the EMP injected by the probe induces a fault in the AES operations. The whole surface of the package was exposed to a localized EMPs with a displacement step of 500μm (which is also the probe diameter).

At each location, an EMP was injected during the last round of the AES and the corresponding faulted ciphertext retrieved. This process was done for 1,000 encryptions of the same plaintext input, and for every of the 30x30 different locations of the injection probe on top the FPGA package.

Figure 11 reports the floorplan of the design (left part) and the associated faults cartography (right part). In this figure, the square in the center corresponds to the FPGA die position. At each location, the number of most frequent faulted bytes are reported.

First we observed that the effect of the EMP is clearly localized in space. Some locations above the surface of the circuit are more sensitive to the EMP than others. When the EMP is localized in the region near the block cipher, the number of faulted bits increases. Second, we observed a good correlation between the most sensitive coordinates (fig. 11) and the position of the ROUNDEXE. This block contains the cipher module depicted in figure 9, which is the place where the critical delay path is located.

C. Fault analysis

Figure 12 shows the behavior of the induced faults for a first random position (X_1,Y_1,Z) on top of the die's surface ($7\text{x}7mm^2$) right in the ROUNDEXE area (the cipher module). 1,000 encryptions were done with random plaintexts and a constant key while injecting EMPs (their parameters are given in fig. 10) during the last round of the AES calculations. The occurrence rates of both mono-bit (i.e. fault affecting a single bit) and multi-bits faults are given in fig. 12.

The path corresponding to the 15^{th} byte appears to be the most sensitive to the EMP at coordinates (X_1,Y_1,Z). For this byte, 3% of the faults were mono-bit, and 80% of the faults were multi-bits faults. It also reveals a data-dependence of the injected faults to the data (in that instance the plaintext) handled by the target. This behavior was corroborated by an inspection of the faults (the fault value is calculated by xoring the correct and corresponding faulty ciphertexts): different faults were obtained for different plaintexts with the same experimental settings. This behavior is as well a feature of faults induced by timing constraints violation (its origin lies in the data-dependence of the data propagation time through combinatorial logic). This is another sign that reinforces the assumption that the fault injection mechanism by means of EMP is related to timing constraints violation.

The same experiment was carried out for two other locations (X_2,Y_2,Z) and (X_3,Y_3,Z) on top of the die with the same 1,000 plaintexts used previously. Figures 13 and 14 report the corresponding mono-bit and multi-bits fault occurrence rates.

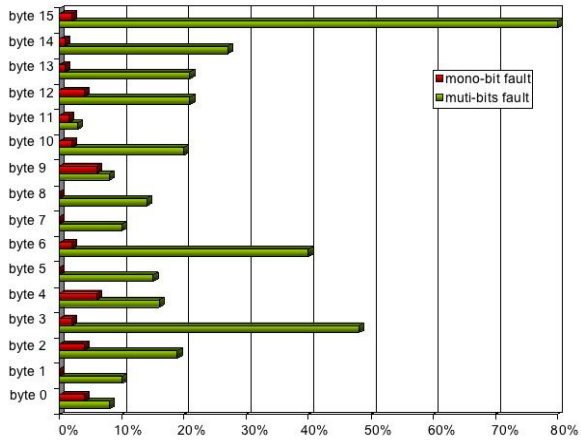

Figure 12. Behavior of the faults at coordinates (X_1,Y_1,Z)

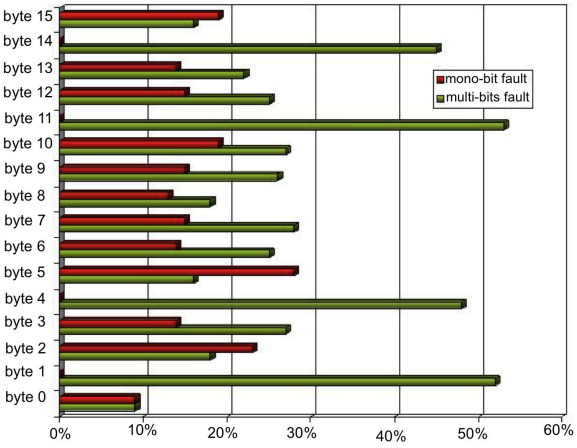

Figure 13. Behavior of the faults at coordinates (X_2,Y_2,Z)

These three figures (12, 13 and 12) exhibit different occurrence rates: the injection probe location has an effect on the induced faults and on their related properties. In fact, at coordinates (X_1,Y_1,Z), the 15^{th} byte is the most sensitive to the EMP. Whereas, at coordinates (X_2,Y_2,Z) and (X_3,Y_3,Z) the most sensitive paths correspond to the 11^{th} byte and to the 7^{th} byte respectively. We observed that the faulted paths were different for different locations of the injection probe. These observations reveal a local effect (i.e. restricted to a part of the device's area) of the EMPs.

The evidence of a local effect demonstrates the ability to fault sub-critical paths. In some locations, the most critical one is never faulted. This is very interesting (for an attacker) since it is possible to select the disturbed path without always

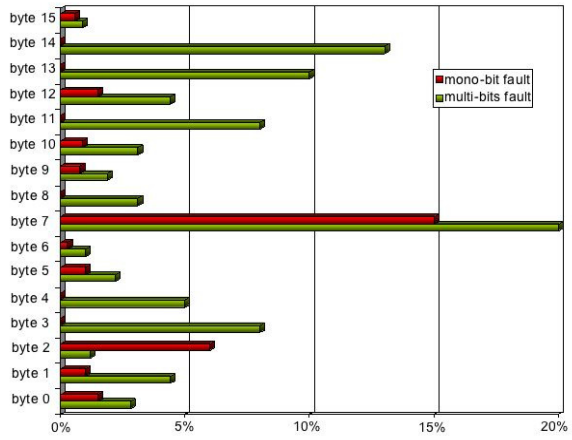

Figure 14. Behavior of the faults at coordinates (X_3,Y_3,Z)

affecting the most critical ones as it is the case for power or clock glitches. These results lead to consider attacks on both data and key schedule paths as described in [14] and [10].

To sum up, we draw the assumption that an explanation of the EMP injection mechanism may lie in a coupling between the EMP and the PGN of the FPGA [20]. This coupling may induce a transient decrease of the voltage applied to the logic of the target. As a consequence, the propagation delays through the logic may have been increased until faults are induced by the violation of the chip's timing constraints. The next section intends to provide more evidences of that assumption.

VI. Transient Electromagnetic Faults on a Hardware Implementation of the AES with Countermeasure

Based on the results obtained in the previous section, we draw the hypothesis that the EMP induces timing violations during AES operations. These timing violations seem to be localized. In order to validate this hypothesis, a countermeasure was added to the previous design whose aim is to detect any timing violations.

In this experiment, the test chip embedded the same hardware 128 bits version of the AES encryption algorithm as described in Section V. The design consists of four main blocks: a communication and control module, a key expansion module, a cipher module and a countermeasure module.

A. Principle of the countermeasure

In CMOS synchronous digital circuits, data are handled by the combinatorial logic and stored in registers synchronized by a common clock (noted CLK). The implemented countermeasure consists in monitoring the data path delay. An alarm is activated when the timing constraints of the circuit are violated, such a violation meaning the appearance of faults.

The countermeasure is based on the insertion of a monitoring delay value ($D_{monitoring}$) between the critical path of the design and the clock period T_{clk}. The principle is illustrated in figure 15. The idea is that any timing violation will be preceded by a violation of the monitoring delay ($T_{clk} < D_{monitoring}$) at the origin of the alarm's activation. Figure 16 illustrates this mechanism in the case of a negative glitch on the power supply voltage of the circuit. For this case, the increase in the propagation delays in the logic comes along with a similar increase in the monitoring delay. The latter being also affected by the power supply glitch. The alarm is then triggered as $D_{monitoring}$ increased over the clock period.

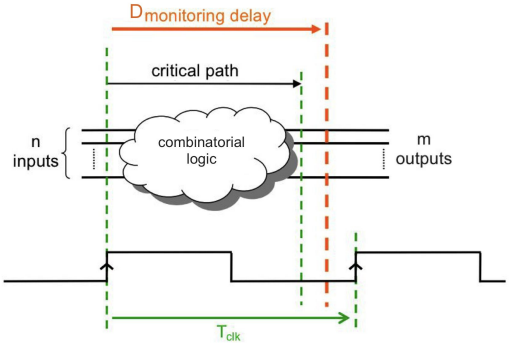

Figure 15. Principle of the countermeasure

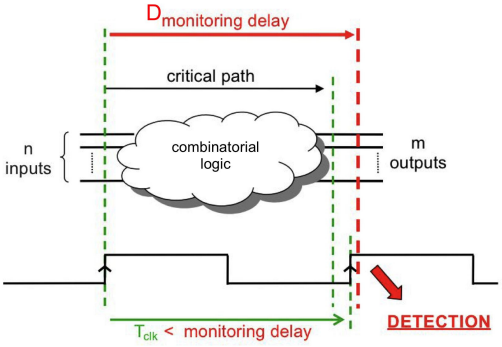

Figure 16. Detection of timing violation

B. Injection of transient EM fault

In order to verify if the countermeasure described above is able to detect the glitch induced by the EMP, we performed an EMP cartography. At each location, an EMP was injected during the last round of the AES, then the faulted ciphertext and the alarm value were stored. This operation

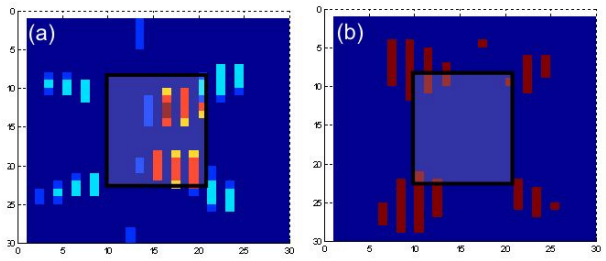

Figure 17. Faults and alarms cartographies

was performed for 1,000 encryptions of the same plaintext input, and for 30x30 different locations on top of the FPGA package. Figure 17 shows the obtained cartographies. At each position, in fig. 17-a the number of most frequent faulted bytes are reported, while in fig. 17-b we report the alarm activation.

C. Faults analysis

From these cartographies, we observed that the effect of the EMP was detected by the countermeasure only in some positions on top of the surface of the circuit. Moreover, in many positions, the EMP has induced a transient fault without being detected by the countermeasure. Considering the 30x30 studied locations of the antenna, the countermeasure has detected the EMP in only 12 positions when the latter has induced faults in 113 positions.

Figure 18 reports the occurrence of the faults and alarms. As we can observe, only 10% of the induced faults triggered the alarm, when 90% of the faults were induced without the triggering of the alarm. Moreover, it seems that in some positions above the surface of the circuit, the alarm is triggered without inducing faults in the calculations. These results confirm the localized effect of the EMP, and moreover the possibility to select the faulted path and not necessary the most critical one; as opposed to the faults induced by clock or power glitches where the critical path is the first path to be faulted.

VII. Conclusion

This paper reports practical injection of transient faults into the calculations of a micro-controller and of a FPGA by means of EMPs. Both targets were embedding the AES encryption algorithm.

Regarding fault injection on the micro-controller, an analysis of the obtained faults revealed that they were induced by skipping the instruction which should have been executed during the EMP. The use of a pulse generator made it possible to change the time of the EMP with a nanosecond accuracy. Consequently, the whole bytes of the AES were faulted independently by modifying the injection time. Every round of the AES may also be targeted. Such an instruction scale accuracy for EMP induced faults was never previously reported. Besides, constant faults (i.e. plaintext independent) were obtained with proper settings. This fault model permit to carry out the fault attack described in [23] which is based on the injection of constant faults. Moreover, the achievement of instruction skipping, as we did, was also reported by Balash et al. in [4] on the same target by means of timing constraints violation obtained with clock glitches. Thus, we drew the assumption that the EMP induced faults were related to timing violation.

Fault injection experiments on an FPGA also revealed the ability to inject single-bit and multi-bits faults into the calculations of the AES. These faults were found data dependent. Moreover, a local effect of EMPs was underlined: the injected faults (if any) are modified when the injection probe location is changed. This property of EMP fault injection is particularly worrying. Indeed, it may allow to bypass many countermeasures intended to prevent fault injection by power supply glitches (e.g. power supply low-pass filtering, use of internal supply monitoring, etc.). This local effect and the assumption that EMP faults are induced by timing constraint violation were further investigated by adding a countermeasure based on monitoring the compliance of timing constraints. It has ascertained both the location dependence of the injected faults and the assumption of an injection mechanism based on timing violation. Consequently, some faults were injected without triggering the alarm. They have affected part of the logic located away from the countermeasure.

Further investigations of the countermeasure area of effectiveness is currently performed. The use of an EMP monitoring matrix spread over the circuit will be also investigate to propose a countermeasure against this emerging threat.

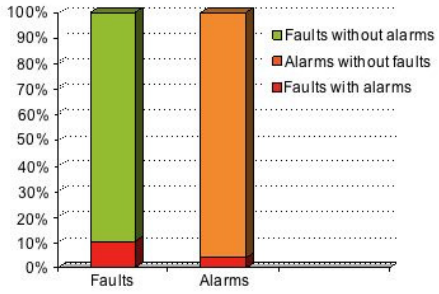

Figure 18. Occurrences of faults and alarms considering the 30x30 coordinates above the package surface

REFERENCES

[1] Agoyan, M., Dutertre, J.M., Mirbaha, A.P., Naccache, D., Ribotta, A.L., Tria, A.: How to flip a bit? In: On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International. pp. 235 – 239 (2010)

[2] Agrawal, D., Archambeault, B., Rao, J.R.: The em side-channel(s):attacks and assessment methodologies. In: CHES (2002)

[3] Alaeldine, A., Ordas, T., Perdriau, R., Maurine, P., Ramdani, M., Torres, L., Drissi, M.: Assessment of the Immunity of Unshielded Multi-Core Integrated Circuits to Near-Field Injection. In: EMC (2009)

[4] Balasch, J., Gierlichs, B., Verbauwhede, I.: An in-depth and black-box characterization of the effects of clock glitches on 8-bit mcus. In: FDTC. pp. 105 –114 (2011)

[5] Barenghi, A., Bertoni, G., Parrinello, E., Pelosi, G.: Low voltage fault attacks on the rsa cryptosystem. In: FDTC. pp. 23–31 (2009)

[6] Bayon, P., Bossuet, L., Aubert, A., Fischer, V., Poucheret, F., Robisson, B., Maurine, P.: Contactless electromagnetic active attack on ring oscillator based true random number generator. In: International Workshop, COSADE 2012, Darmstadt, Germany, May 3-4, 2012. Proceedings. Lecture Notes in Computer Science, Springer

[7] Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: CRYPTO. pp. 513–525 (1997)

[8] Blömer, J., Krummel, V.: Fault based collision attacks on aes. In: FDTC. pp. 106–120 (2006)

[9] Bruestle, M.: sosse - simple operating system for smartcard education (2002), `http://www.mbsks.franken.de/sosse/index.html`

[10] Chen, C.N., Yen, S.M.: Differential fault analysis on aes key schedule and some countermeasures. In: Proceedings of the 8th Australasian conference on Information security and privacy. pp. 118–129 (2003)

[11] Choukri, H., Tunstall, M.: Round reduction using faults. FDTC pp. 13–24 (2005)

[12] Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: CHES (2001)

[13] Giraud, C.: Dfa on aes. Advanced Encryption Standard - AES, 4TH International Conference 3373, 27–41 (2003)

[14] Giraud, C.: DFA on AES Advanced Encryption Standard – AES. Lecture Notes in Computer Science, vol. 3373, p. 571. Springer Berlin / Heidelberg (2004)

[15] Hamid, H.B.E., Choukri, H., Tunstall, M., Naccache, D., Whelan, C.: The sorcerer's apprentice guide to fault attacks 94 (2004)

[16] Kim, C., Quisquater, J.J.: Fault attacks for crt based rsa: New attacks, new results, and new countermeasures. In: Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems. Lecture Notes in Computer Science, vol. 4462, pp. 215–228. Springer Berlin / Heidelberg (2007)

[17] Li, Y., Sakiyama, K., Gomisawa, S., Fukunaga, T., Takahashi, J., Ohta, K.: Fault sensitivity analysis. In: Mangard, S., Standaert, F.X. (eds.) CHES, Lecture Notes in Computer Science, vol. 6225, pp. 320–334. Springer Berlin / Heidelberg (2010)

[18] NIST: Announcing the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication, n. 197 (Nov 26, 2001)

[19] Piret, G., Quisquater, J.J.: A differential fault attack technique against spn structures, with application to the aes and khazad. In: CHES. pp. 77–88 (2003)

[20] Poucheret, F., Tobich, K., Lisart, M., Robisson, B., Chusseau, L., Maurine, P.: Local and direct em injection of power into cmos integrated circuits. In: FDTC (2011)

[21] Quisquater, J.J., Samyde, D.: a new tool for non-intrusive analysis of smartcards based on em emissions. In: Rump Session Eurocrypt (2000)

[22] Quisquater, J.J., Samyde, D.: Eddy current for Magnetic Analysis with Active Sensor. In: Esmart (2002)

[23] Roche, T., Lomne, V., Khalfallah, K.: Combined fault and side-channel attack on protected implementations of aes. In: CARDIS (2011)

[24] Schmidt, J.M., Herbst, C.: A practical fault attack on square and multiply. In: FDTC. pp. 53–58 (2008)

[25] Schmidt, J.m., Hutter, M.: Optical and em fault-attacks on crt-based rsa: Concrete results. In: Austrochip. pp. 61–67 (2007)

[26] Selmane, N., Guilley, S., Danger, J.L.: Practical setup time violation attacks on aes. In: EDCC. pp. 91–96. IEEE Computer Society (2008)

[27] Skorobogatov, S.P., Anderson, R.J.: Optical fault induction attacks 2523, 2–12 (2002)

[28] Van Woudenberg, J.G., Witteman, M.F., Menarini, F.: Practical optical fault injection on secure microcontrollers. In: FDTC. pp. 91 –99 (2011)

[29] Yen, S.M., Joye, M.: Checking before output may not be enough against fault-based cryptanalysis. IEEE Transactions on Computers 49(9), 967–970 (2000)