Single-bit DFA using multiple-byte laser fault injection

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

Authors' version
Technologies for Homeland Security (HST), 2010
DOI : 10.1109/THS.2010.5655079

Single-Bit DFA Using Multiple-Byte Laser Fault Injection

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*—Laser fault injection is known as a fault attack method on cryptographic systems. This work provides practical experiments on an 8-bit 0.35μm microcontroller with no countermeasures. It explains how, with a laser beam that creates multiple-byte faults, it is still possible to perform single-bit/byte Differential Fault Analysis (DFA). It requires spatial and temporal adjustments for laser shooting and faulty results classification. This underlines the need to protect cryptographic devices, such as biometric passports and smart cards against surgical faults targeting one or several single-bits on specific bytes in memory.**

I. Introduction

Cryptosystems contain secret keys for cryptographic algorithms used to protect confidential information or to authentication processes. For this reason, they are always the subject of much research aimed at improving their security and resistance to any unauthorized interference.

To break cryptographic systems, attacks are usually assigned into two categories: *mathematical* and *hardware*. The second category targets the physical implementation of the cryptographic algorithms. It consists on different types of attacks and among them the *fault attacks*, based on the analysis of correct and faulty encryption results or hardware malfunctioning.

Advanced Encryption Standard (AES) has been promulgated and endorsed on November 2001 by the Federal Information Processing Standard (FIPS) as an approved secret key cryptographic algorithm for protecting unclassified sensitive electronic data. Besides, on June 2003, US National Security Agency (NSA) has accepted to use AES for secret and top secret information under the conditions of sufficient key length and prior approval of the implementation [19].

We report in this paper the results of our security characterization study on a microcontroller that runs an AES. Our research concerns retrieving the secret key of the AES by laser fault attacks. For this aim, we undertake to inject very limited laser faults that could satisfy a required fault attack model.

Nowadays, as the technology progresses, the density of transistors by area unit increases. However, the minimal diameter of a laser spot could not yet successfully decrease to smaller than 1μm due to optical diffraction reasons. This small laser spot needs very accurate and expensive advanced equipments and it is not available for most opponents. Even this minimal beam hits several transistors on new technologies and can not physically be limited to target a single-bit or even a single-byte of memory. So, the possibility of injecting limited laser faults that satisfy a required model is sometimes considered very low. Therefore, is there still a threat from single-bit/byte laser fault injection for current and ongoing technologies?

To achieve our aims, we take advantage of using a laser bench. We will show that it is possible to achieve a multi-byte fault attack that permits to perform a single-bit/byte analysis model to discover the key. It requires setting up some parameters related to the laser and adjusting spatial and temporal coordinates. Then, it is also necessary to perform a classification of the faulty results. The consequences are that appropriated countermeasures must be designed, implemented and evaluated to protect future cryptographic systems against possible fault injection procedures.

In the next sections, after describing briefly the cryptographic attacks, the AES algorithm and laser fault injection, we will discuss about our tests and our results.

II. Attacks On Cryptographic Circuits

Attacks on cryptographic circuits can be categorized into two main families:

A. Cryptanalytic or Mathematical Attacks

These attacks search for vulnerabilities in a cryptographic schema or algorithm to deduct the keys by mathematical methods [14]. A brute-force search for the key can also be considered as a cryptanalytic attack.

The key length of reliable cryptographic algorithms increases continually above the progresses of calculation capability of computers for finding the keys. So, a brute-force search for their keys cannot give any answer in a reasonable amount of time, except if it has been applied as a complement of another attack that can reveal a great part of a key.

B. Hardware Attacks

This large family of attacks targets hardware components and includes three sub-categories:

1) Side Channel Analysis: These attacks are based on the analysis of any information leakage from a circuit during the encryption operations, related to sensitive data processing that can reveal the secret key. It consists on different types of measurement, such as power consumption, electromagnetic radiation, heat emission or response time of a circuit [13].

2) Invasive Attacks: This subcategory covers all the techniques based on the analysis or modification of an IC's design by an invasive method, such as abrasion or chemical etching. It includes also the use of focused ion beams for changing the chip's interconnections [15].

3) Fault Attacks: This last type of hardware attacks is based on intentionally modifying a chip's environment to alter its functioning [24]. It can be performed in different manners:

- Changing the normal behavior of a circuit, modifying the machine state or reducing the round number and then exploiting the differences. These kinds of attacks can also be subcategorized under *Differential Behavioral Analysis (*DBA*)* [22], *Round Reduction* [17], etc.
- Gaining some insights into the secret data handled by the circuit and finding the secret key by comparing a faulty and its corresponding correct ciphertext. These analysis techniques are known as *Differential Fault Analysis (*DFA*)* [5] [3].

In both cases, the faults are induced into the circuit through the use of different means, such as increasing the temperature, exposing to laser, UV or X radiations or intense pulsed light (*e.g.* a camera flash) or modifying clock frequency. For more information, please refer to [24].

Our experiments are based on using laser fault injection and DFA methods for retrieving AES key.

C. Symmetric Cryptography and AES

Modern cryptography includes symmetric and asymmetric methods. In the first family, messages are encrypted using a unique secret key that provides the security for the sender and the receiver.

AES is a symmetric method and is based on Rijndael cipher [19]. It can grant a high level security using a reasonable calculation time. So, AES was quickly adopted for many systems and products after FIPS validation in 2001. Thus, many types of attacks have been studied by researchers with the intention of improving AES implementations by suitable countermeasures.

On June 2003, US National Security Agency (NSA) has announced that *"The design and strength of all key lengths of the* AES *algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the* SECRET *level.* TOP SECRET *information will require use of either the 192 or 256 key lengths"* [8]. However, it noticed that *"The implementation of* AES *in products intended to protect national security systems and/or information must be reviewed and certified by* NSA *prior to their acquisition and use"* [8].

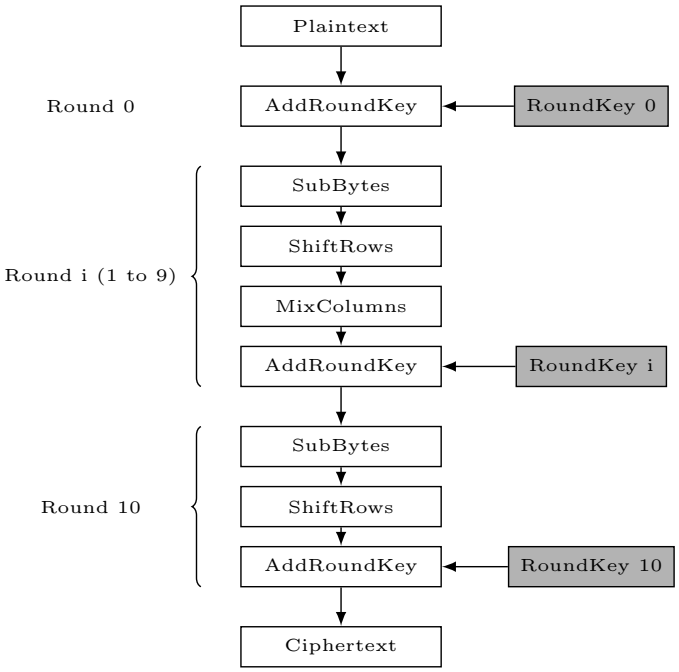

Fig. 1. The AES - General Outline.

Therefore, detection and mitigation of any potential threat is very substantial for AES systems security. Today, a significant part of researches in cryptography is focused on improving them against any eventual attack.

AES is an algorithm that performs message encryption processing by data blocks of 128 bits at input and output using a key size of 128, 192 or 256 bits respectively in 10, 12 or 14 rounds (after a short initial round) according to the size of the key. The algorithm includes two separated processes: One for the key scheduling to derive the round keys from the secret key and another one for the data encryption.

Decryption also is divided into two separated processes: One for the inverse key scheduling and another one for the data decryption.

For the initial round in AES-128, the algorithm uses the secret key as the round key; but for each following round, the corresponding round key is calculated from the previous one. Figure 1 shows the different operations of the AES-128 algorithm. Hereafter, we use AES to refer to AES-128 and we use the "K" prefix plus the number of a round to refer to a round key (*e.g.* "K_9" for the round key of the 9-th round).

To encrypt a plaintext, namely M, according to the implementation of AES, usually at the beginning of algorithm execution, all the round keys are computed from the main key and are stored in the memory. Then, the encryption process begins and takes separated blocks of 16 bytes (128 bits) from M as input and puts each block in a matrix of 4×4 bytes.

Each round of the algorithm, except the initial and the last ones, includes 4 steps: At the beginning, it exchanges the value of each matrix element by the corresponding value in a substitution table (`SubBytes` or `SB`). Then, it executes a rotational operation on the matrix rows (`ShiftRows` or `SR`). In the third step, the algorithm applies a linear transformation to each element and combines it with other values of the same column with a different coefficient of 1, 2 or 3 for each element (`MixColumns` or `MC`) under the specific rules of GF(2^8). This step guarantees the distribution of the information of each byte on 4 bytes and increases security of encrypted messages. In the last step of each round, a bitwise xor operation is performed between the value of each element and the corresponding byte on the round key (`AddRoundKey` or `ARK`).

Currently, AES encryption is widely used for governmental, military and commercials purposes. Therefore, it has opened a new and large domain of research on security of cryptographic circuits.

III. Laser Fault Injection

Laser (Light Amplification by Stimulated Emission of Radiation) is an emitted electromagnetic radiation as visible or invisible light amplified by stimulated emission. A laser light is monochromatic, unidirectional, coherent and undiscoverable in nature. It can be generated as a beam with a very small diameter that is only micrometers width, and then conducted, even through materials to perturb a target in a very restricted time.

A conducted laser beam on a circuit's component can alter its functioning. For example, an SRAM (Static Random Access Memory) cell storing an information bit, exposed to a laser shoot may flip to the opposite value, leading to a fault injection. This phenomenon is known under the term of *Single Event Upset (*SEU*)*. It is induced by voltage transients due to the photoelectric effect of a laser beam on CMOS logic [16] [25].

By setting up the energy level below a destructive threshold value, laser attack will not damage the device. Please refer to [23] [9] [7] for more information.

Several parameters are at stake in every laser attack on a circuit, especially laser spot size (or diameter), wavelength, amount of emitted energy value and exposure duration. Depending to the laser bench technology and facilities, the opponent may take over the control some of these parameters.

Moreover, each side of a chip has different characteristics when a laser beam hits its surface:

1- The front side shows a good visibility of the layout. But accurate targeting of a multi-metal layers component on it is difficult because of the reflective effect of metallic interconnects. In addition, as the fabrication technology advances, the number of metal interconnects on a chip area grows and its size reduces. So, it becomes more and more difficult to reach the proper area on front side of the chips in ongoing technologies.

2- The backside does not provide any visibility of the layout. So, the positioning is more difficult, except by using more special equipments, such as an infrared camera. In addition, an infrared wavelength (~ 1064nm) is necessary for the laser beam to enter deeply in silicon and to alter the sensitive areas. Although, as the reflective problem of metallic surfaces no more exists, a laser attack on the backside is almost more efficient.

IV. Giraud's Bit DFA by Laser

DFA methodology consists of analyzing the encryption results after injecting faults, according to a required model of fault attacks. The faulty results are then compared with their corresponding correct ciphertexts to extract information by cryptanalysis.

Several bit-level or at byte-level DFA models on AES exist (*e.g.* [12] [18] [10] [4]). According to AES algorithm, by finding only one of the round keys, the secret key may be calculated. Although, these DFA models are sometimes considered as very impractical or even infeasible, especially for new and ongoing technologies.

Christophe Giraud has developed two efficient DFA methods at bit and at byte levels to retrieve the secret key of an AES [12]. His bit-level method requires to inject a single-bit fault before the `SubBytes` input of the final round and searches to retrieve K_{10} value. For discovering successfully one byte of K_{10}, the opponent needs to generate single-bit faults for at least three different plaintexts and then to compare correct and faulty results of each text.

Giraud's single-bit attack is not the most effective attack on AES. There are other attacks that are more performant and even they could discover the secret key using just two texts, such as [20]. But, Giraud's bit method seems to be the most difficult to implement, as this attack needs to change only one bit on specific bytes.

This method benefits from the properties of xor operations: An xor operation between the correct ciphertext, namely C and the faulty ciphertext, namely D, calculates the Δ value that shows differences between them. Then, the Δ value conduct the opponent to a set of assumptions for the corresponding value of K_{10}.

In a normal processing, the value of each byte in the ciphertext is calculated by an xor operation between its corresponding value on K_{10} and `SubBytes` of corresponding value at the output of the 9-th round (M_9):

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

For more clearness, hereafter we write any equation according to the value of discrete bytes. We omit the `ShiftRows` operation that has not any effect on the byte contents. So, (1) can be rewritten as (2):

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

When a single-bit fault e is injected on the `SubBytes` input of the 10-th round, faulty ciphertext can be written as (3):

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

Then, the difference between correct and faulty values (Δ) can be calculated by an xor operation between them, as shown in (4):

$$\Delta = C \oplus D \tag{4}$$

As the Δ value is known, (4) can be rewritten as (5). The new equation conducts the opponent to create a set of assumptions for the values of M_9 and e:

$$\Delta = \mathrm{SB}(M_9 \oplus e) \oplus \mathrm{SB}(M_9) \tag{5}$$

Although, by using (6), derived from (3), the set of assumptions can be created on the value of a K_{10} byte:

$$K_{10} = \mathrm{SB}(M_9 \oplus e) \oplus D \tag{6}$$

By repeating the fault injection for each different text, the opponent can create a new set of assumptions for the value of corresponding K_{10} byte. Finally, the value of one K_{10} byte can be retrieved by determining the intersection of the assumption sets.

With a probability of about 97%, three plaintexts suffice to discover a byte of K_{10} [12]. Otherwise, the opponent iterates the process for more plaintexts to until the sets' intersection reaches a singleton.

By repeating this procedure for 15 other bytes, K_{10} can be entirely retrieved. Afterwards, the opponent can calculate the secret key by reversing the key scheduling processes.

V. Practical Multiple-Bit Fault Injection

For our tests, we used an 8-bit 0.35μm 16 MHz RISC microcontroller with an integrated SRAM of 4 KBytes and no countermeasures. The device runs SOSSE (Simple Operating System for Smartcard Education) [6] to which we added some commands, most notably for feeding-in plaintexts and retrieving ciphertexts coded by AES. A 128-bit K was embedded in the code. As encryption starts, the K_is are derived and stored in SRAM.

Our test bench was equipped with a YAG (Yttrium Aluminium Garnet or $Y_3Al_5O_{12}$) laser emitter in three different wavelengths: Green, infrared and ultraviolet. The nominal spot diameter could be set between 0 and 2500 μm. As the beam passes through a lens, it gets reduced by the lens' zoom factor and it loses a big part of its energy. Our experiments were conducted with a $20\times$ Mitutoyo lens, a green beam ($\sim$ 532nm) of about $\varnothing 5.5\mu$m and $\simeq 15pJ$ per shot [1]. The circuit was fixed on a motorized and programmable X-Y positioning table for upright microscopes with 0.1 μm steps. 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.

[1] At the laser source emitter, before passing through the lens.

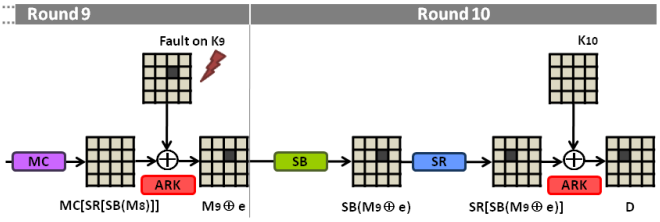

Fig. 2. Consequences of a single-bit/byte injected fault on K_9.

The chip was decapsulated by chemical etching using a Nisene JetEtch automated acid decapsulator. For opening our chips, we used only nitric acid (HNO_3) at $80°$C for 40 seconds. The etched chip successfully passed the functional tests before and during fault injection.

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

To perform Giraud's bit attack, the opponent needs to inject a single-bit fault before the `SubBytes` input of the last round. A means to meet this requirement is to inject a single-bit fault on K_9 that results consequently in such a fault on M_9 through the `AddRoundKey` operation. Figure 2 shows the consequences of an injected single-bit fault on K_9 through the 10-th round on the temporary ciphertext and on the final ciphertext. In our implementation, at the beginning of operations, all the round keys are calculated from the main key and are stored in the SRAM.

Finding corresponding area and proper beam parameters is the most important step for successfully performing the tests, but it is also very time consuming. The number of faults on the ciphertext, their position and their content are very significant to understand which round key has been attacked.

In previous papers [1] [11], we showed that a reproducible single-bit or a single-byte fault injection by a laser spot that hits few more bytes is possible. We described that by conducting an accurate temporal laser fault attack, the opponent can discard logically the effect of few faulty bytes that appear on previous round keys. He can deceive the encryption process to use only a single-bit or a single-byte fault (between several faults that exist physically) during encryption.

The minimal diameter of a laser spot could not yet successfully decrease to smaller than $\varnothing 1\mu$m due to optical diffraction reasons. As the technology advances, the number of transistors grows on the incident area of a $\varnothing 1\mu$m spot, so single-bit/byte fault injection will need more accurate equipments and becomes less feasible by cheap laser facilities. In addition, a $\varnothing 1\mu$m spot may have a bigger effective area on the chip that depends to the laser energy level. This minimal beam hits several transistors on new technologies and can not physically

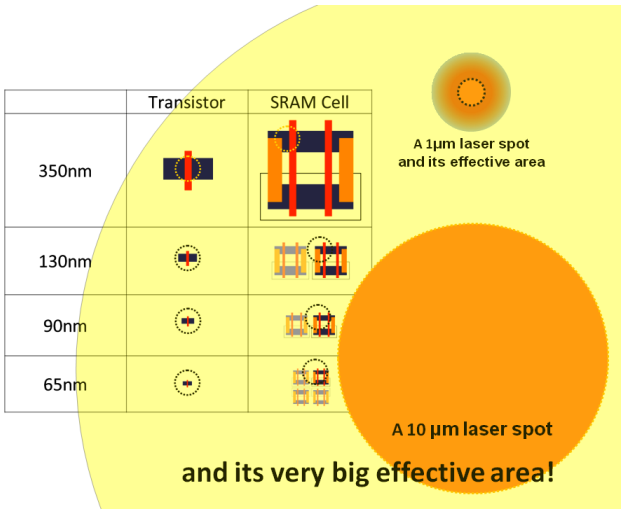

Fig. 3. 1 μm & 10 μm laser spot diameters vs technology scaling.

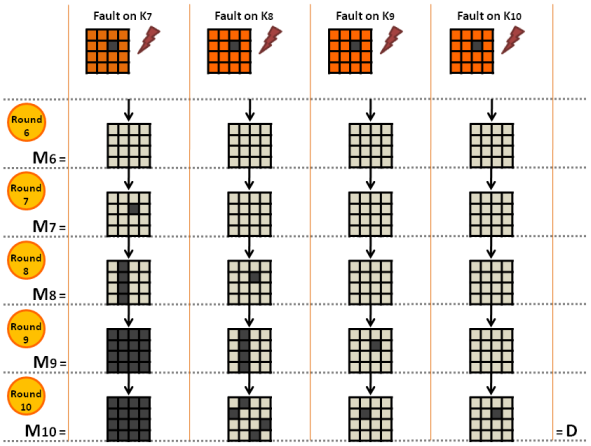

Fig. 4. Effects of `MixColumns` for different single-bit/byte faulty round keys on the temporary ciphertext at the end of each round and at the end of algorithm.

limited to target a single-bit/byte. However, a laser equipment providing with $\varnothing 1\mu$m spot is very expensive and not accessible to most of opponents. So, is it possible to use a bigger laser spot to perform such attacks? By example, by using a $\varnothing 5$ or 10μm laser spot, is it still possible to perform single-bit/byte DFA? Figure 3 shows a comparison between $\varnothing 1$ and 10μm laser spot and an SRAM cell in different technology scaling.

As an AES algorithm has a `MixColumns` step at each round, except at the initial and final rounds, a single-bit/byte fault injected before the input of any round alters 4 bytes of the temporary ciphertext at the end of the round. So, a single-bit/byte fault on K_9 or on K_{10} that don't pass through any `MixColumns` step, changes only one byte on the ciphertext, as shown in Figure 4. While a single-bit/byte fault on K_8 alters 4 bytes of the ciphertext and a similar fault injected on any round key before K_8 will fault the whole bytes of the ciphertext.

However, injected faults are not usually limited to a single byte and/or a single round key. When more than 4 bytes are faulty on a ciphertext, it is difficult to understand if they correspond to an injected fault on any round key before K_8 or many injected faults on K_8, K_9 and K_{10}.

In several physical implementations of SRAM (*e.g.* in our microcontroller or in [23]), it seems that the bits of a same value are designed and built close together for a block of bytes in the memory array. In theses implementations, usually the distance of two bit cells of same value in a block of bytes (*e.g.* 256 bytes) is much closer than the distance of a bit with its neighbor bits of the same byte. This is a weakness point for the security of SRAM contents against single-bit fault injection. As the SRAM of our chosen microcontroller has this weakness, we could inject successfully single-bit faults on several bytes with a laser beam.

To perform our tests, we tried to inject at least one single-bit fault on K_9 and to protect K_{10} from any fault. In this case, several faults are injected on previous round keys, but

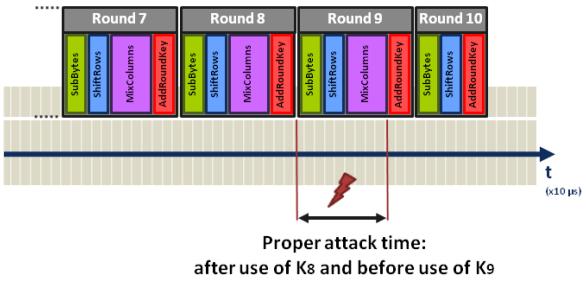

Fig. 5. Proper temporal localization for fault injection on K_9 in a time period that previous round keys are used. So, effect of any fault on previous round keys is discarded from the ciphertext.

by a temporal accuracy, their effects are discarded from the encryption process. As we described in previous works [1] [11], we performed the laser attacks after the use of K_8 and before `AddRoundKey` of K_9 (Figure 5).

The figures 6 and 7 show SRAM bytes that contain round keys values. Each small case represents one byte. Those are just simple representations of SRAM memory and don't correspond to real physical implementations of the memory or address allocations. However, they can demonstrate a simple logical model of our attack.

As shown on the Figure 6, this attack results in one or several single-bit faults on K_9. If any fault is not injected on K_{10}, the faulty ciphertext permits to perform Giraud's bit DFA. As there is not any `MixColumns` step in the final round, each faulty byte on the ciphertext is independent from other bytes. This is also an advantage to reduce the number of laser fault injection attacks to have the required number of pairs of correct/faulty ciphertext of same plaintext for each byte.

Figure 6 shows how the opponent can limit the single-bit fault injection to one or several byte of K_9 by controlling the laser shooting time. However, it is not sometimes possible to

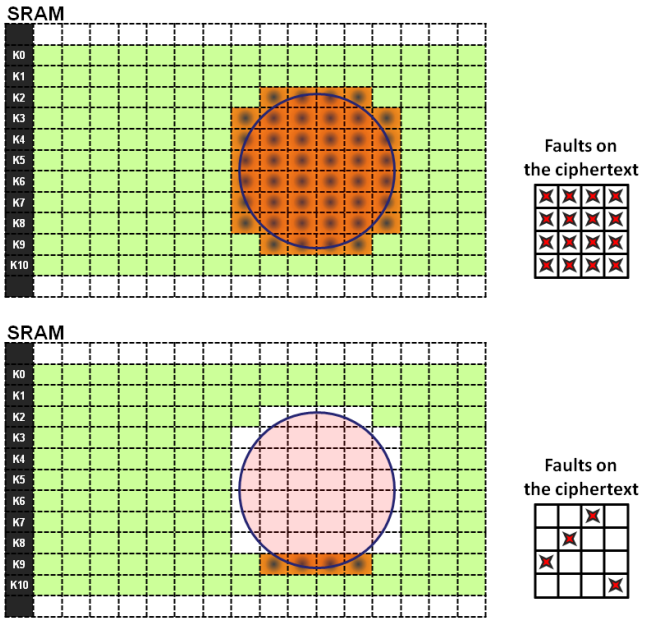

Fig. 6. Spatial localization for laser fault injection on K_9, before and after a temporal adjustment that discards the logical effect of any fault on previous round keys.

protect K_{10} from any fault injection. So, we can consider an advanced case, when the laser beam hits few bytes on K_{10} (Figure 7).

In this case, the faulty bytes on K_{10} correspond again to a single faulty bit (regarding to the described memory array physical implementation). But, as there is not any other step after `AddRoundKey` of 10-th round, they create only single-bit change on any corresponding byte of the ciphertext. So, these faulty bytes on the ciphertext due a faulty K_{10} byte have only a single-bit difference with their correct values.

Consequently, for the faulty K_{10} bytes, D is calculated as (7) :

$$D = \mathtt{SB}(M_9) \oplus (K_{10} \oplus e) \tag{7}$$

Here, Δ shows a single-bit difference that corresponds to the injected fault e in K_{10}:

$$\Delta = C \oplus D = e \tag{8}$$

According to (2), (7) and (8), for any faulty byte on the ciphertext, if Δ shows a single-bit difference between C and D, the faulty key byte comes from K_{10}, else it comes from K_9.

Therefore, by using (8), the opponent can classify the faulty bytes on the ciphertext to "K_9 and K_{10} *-related*" fault classes. The class of K_9-related fault refers to all the faulty bytes with more than one bit differences in comparison to their correct value. Although, the class of K_{10}-related fault contains all the faulty bytes with only one bit difference. As the contents of this class don't correspond to Giraud's bit fault model, they

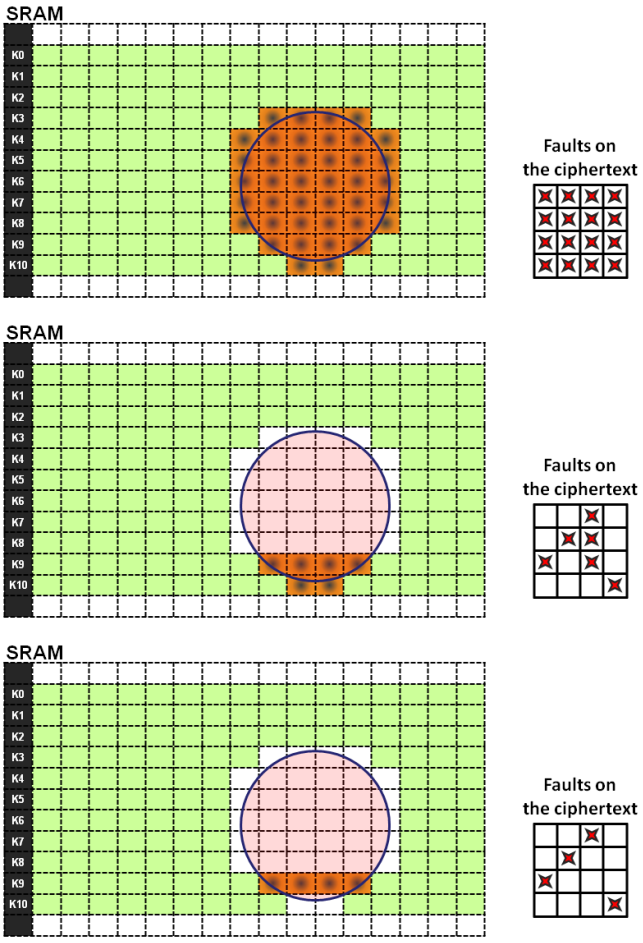

Fig. 7. Spatial and temporal localization for laser fault injection on K_9. At the end, by performing a classification on the faults, two faults caused by faulty K_{10} bytes are excluded from corresponding faults to Giraud's bit DFA model.

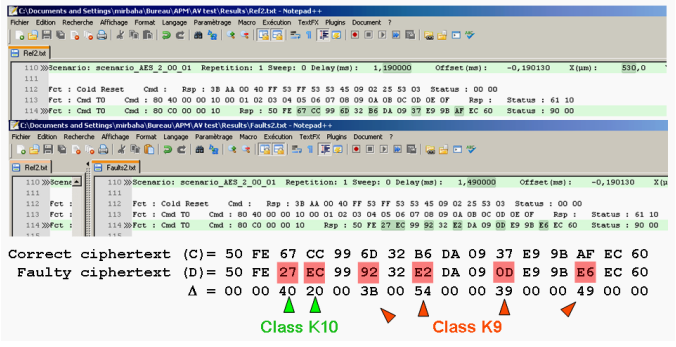

Fig. 8. Classification of faulty bytes on the ciphertext. Faults are separated into two classes of corresponding (K_9-related) and non corresponding (K_{10}-related) faults for Giraud's bit DFA.

must be excluded from list of faulty bytes for the DFA. So, the opponent uses just the class of K_9-related faults to make the assumptions on K_{10} values (Figure 8).

However, two other cases are also possible:

1- A faulty byte on the ciphertext is the result of one

faulty byte on K_9 and another one on K_{10}. In this case, (8) can not discover the effect of K_{10} and it will be classified as a K_9-related fault. So, it creates some false assumptions on K_{10} values. Thus, by intersection operations, the false assumptions will be discarded from the set. However, usually an additional pair of faulty and its corresponding correct ciphertext is needed to reduce the number of assumptions to a single one. Or, an exhaustive search will be needed for the remaining assumptions to examine them and find the correct one.

2- A faulty byte on K_9 creates only a single-bit difference as fault on its corresponding value on the ciphertext. In this very exceptional case, the faulty byte will be classified by error in the class of K_{10}-related faults. So, an additional pair of faulty and its corresponding correct ciphertexts will be needed to reduce the number of assumptions to a single one. Or, like the other case, an exhaustive search will be needed for the remaining assumptions to examine them and find the correct one.

Therefore, we can perform successfully Giraud's DFA using a limited set of faults that correspond to K_9 faults and omit other faults that exist physically on previous and next round keys. This is the exact assumption of Giraud's scenario for single-bit fault injection.

VI. Conclusion

We implemented a Giraud's single-bit attack [12] using laser fault injection. In summary, this note's main conclusions are:

- When the laser beam encounters several bytes, spatial and temporal accuracy can discard the effects of injected faults on previous round keys of K_9. In this case, several single-bit faulty bytes on K_9 increase the performance of Giraud's bit DFA.
- When the temporal accuracy can not protect K_{10} from laser fault injection, a classification between K_9 and K_{10} -related faults may exclude the faults of second class from the DFA process.
- The reproducible single-bit fault injection by big laser spots and Giraud's bit DFA are more feasible than they are usually considered on unprotected chips. So, developing the proper countermeasures against laser fault attacks is necessary for the security of cryptographic circuits.

References

[1] M. Agoyan, J.M. Dutertre, A.P. Mirbaha, D. Naccache, A.L. Ribotta and A. Tria, *How to flip a bit?*, International On-Line Testing Symposium – Proceedings of IOLTS 2010, IEEE, 2010, pp. 235–239.

[2] 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.

[3] 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.

[4] 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.

[5] D. Boneh, R. DeMillo and R. Lipton, *On the importance of checking ryptographic protocols for faults*, Advances in Cryptology – Proceedings of EUROCRYPT 97, LNCS, vol. 1233 , Springer-Verlag, 1997, pp. 37–51.

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

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

[8] Committee on National Security Systems (CNSS), *National policy on the use of the advanced encryption standard (AES) to protect national security systems and national security information*, CNSS Policy (15), 2003.

[9] 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.

[10] 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.

[11] J.M. Dutertre, A.P. Mirbaha, D. Naccache and A. Tria, *Reproducible single-byte fault injection*, Conference on Ph.D. Research In Microelectronics & Electronics – Proceedings of PRIME 2010, IEEE, 2010, In press.

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

[13] M. Joye and F. Olivier, *Side Channel Analysis*, Encylcopedia of Cryptography and Security , Kluwer Academic Publishers, 2005, pp. 571–576.

[14] I. Khaled Salah, A. Darwish and S. Oqeili, *Mathematical attacks on RSA cryptosystem*, Journal of Computer Science, vol. 2 (8), Science Publications, 2006, pp. 665–671.

[15] O. Kommerling and M.G. Kuhn, *Design principles for tamper-resistant smartcard processors*, Workshop on Smartcard Technology – Proceedings of WOST 1999, USENIX Association, 1999, pp. 9–20.

[16] D. Lewis, V. Pouget, F. Beaudoin, Ph. Perdu, H. Lapuyade, P. Fouillat and A. Touboul, *Backside laser testing of ICs for SET sensitivity evaluation*, IEEE Transactions on Nuclear Science, vol. 48 (6), IEEE, 2001, pp. 2193–2201.

[17] Y. Monnet, M. Renaudin, R. Leveugle, Ch. Clavier and P. Moitrel, *Case study of a fault attack on asynchronous DES crypto-processors*, Workshop on Fault Diagnosis and Tolerance in Cryptography – Proceedings of FDTC 2006, LNCS, vol. 4236, Springer-Verlag, 2006, pp. 88–97.

[18] 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.

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

[20] 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.

[21] 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

[22] B. Robisson and P. Manet, *Differential behavioral analysis*, Cryptographic Hardware and Embedded Systems – Proceedings of CHES 2007, LNCS, vol. 4727, Springer-Verlag, 2007, pp. 413–426.

[23] 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.

[24] 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

[25] F. Wang and V. Agrawal, *Single event upset: An embedded tutorial*, Proceedings of International Conference on VLSI Design, IEEE, 2008, pp. 429 –434.