Reproducible Single-Byte Laser Fault Injection

Jean-Max Dutertre, Amir-Pasha Mirbaha, David Naccache, Assia Tria

Reproducible Single-Byte Laser Fault Injection

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

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

I. Introduction

Fault attacks consist in using hardware malfunction to infer secrets from the target's faulty outputs. Within fault attacks, Differential Fault Analysis [2] (DFA) is a particular analysis technique exploiting differences between correct and faulty outputs. We refer the reader to [13] for more information on fault injection techniques.

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

II. The Advanced Encryption Standard

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

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

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

In most implementations the K_is are computed and stored in memory before encryption starts. Encryption treats the 16-byte plaintext M as a 4×4 byte matrix. Each round, except the first and the last, includes four steps: A substitution of the matrix's contents using a lookup table (`SubBytes`), a rotation of the matrix's rows (`ShiftRows`) and a linear transform in GF(2^8) (`MixColumns`) combining each matrix element with other column elements weighted by different coefficients (1, 2 or 3). At the end of each round K_i is xored with `MixColumns`'s result (an operation called `AddRoundKey`).

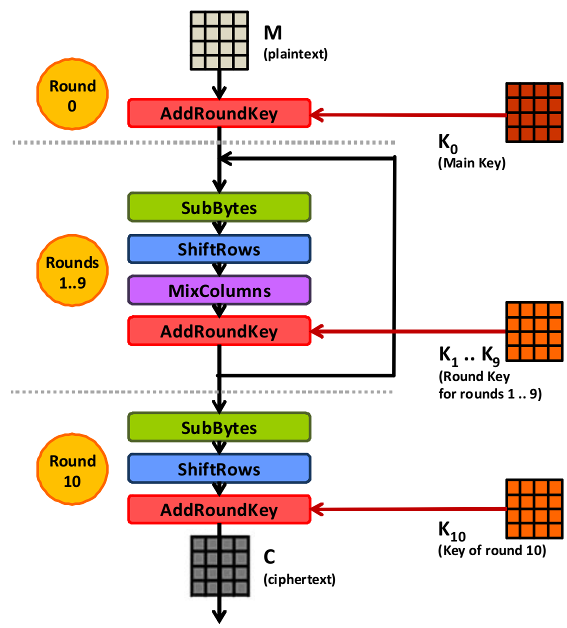

Fig. 1. The AES - General Outline.

III. Laser Fault Injection

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

Laser impacts on electronic circuits are known to alter functioning. In particular, SRAM (Static Random Access Memory) laser exposure is known to cause bit-flips [12], [6], [5], a phenomenon called *Single Event Upset* (SEU). By tuning the

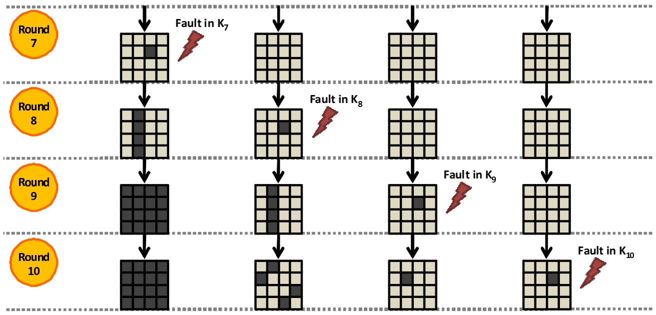

Fig. 2. Effects of one faulty round key byte at different rounds.

beam's energy level below a destructive threshold, the target will not suffer any permanent damage. In a laser attack, the opponent usually controls the beam's diameter, wavelength, amount of emitted energy, impact coordinates (attacked circuit part) and the exposure's duration. Sometimes, the opponent may also control the impact's moment[1], the target's clock frequency, V_{cc} and temperature. Finally, laser attacks may indifferently target the chip's front side or back side.

IV. Piret-Quisquater DFA

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

[11] describes a byte-level DFA on AES (the so-called Piret-Quisquater's DFA). This attack requires the injection of a single-byte fault into the temporary ciphertext between the `MixColumn` exit of the antepenultimate round and the `MixColumn` input of the penultimate round to be successful. A means to meet this requirement is to inject a single-byte fault into the antepenultimate round key (namely K_8). As a consequence, a faulty ciphertext with four faulty bytes is obtained (see Figure 2). Then, the attack scheme described in [11] allows to infer some informations on the four corresponding bytes of K_{10} by processing the correct and faulty ciphertexts and checking over the list of all the related possible single-byte faults.

By repeating this process twice (i.e. by iterating the attack for a different plaintext) the exact value of the four bytes of K_{10} is found with a success rate of about 98% [11]. The procedure is repeated to target K_{10}'s remaining bytes. Finally, $K = K_0$ is inferred by reversing the key schedule. We show that this attack can be implemented, even when the laser spot is wider than the SRAM's cell.

V. Practical Single-Byte Fault Injection

Outline: After chip decapsulation and a mapping of the chip's components, we selected a large target area, given our

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

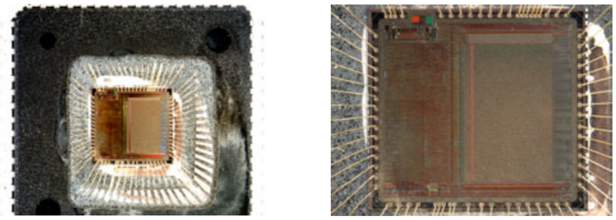

Fig. 3. Decapsulated chip (SRAM is on the left middle and bottom side).

knowledge of the implementation. Using automated search on the chip's front-side, we modified the impact's coordinates, the beam parameters and timing until a one byte fault was obtained. Finally, Piret-Quisquater's attack was performed.

The target is an 8-bit 0.35 μm 16 MHz RISC microcontroller with an integrated 4KB SRAM and no countermeasures. The device runs SOSSE (Simple Operating System for Smartcard Education [4]) to which we added some commands, most notably for feeding-in cleartexts and retrieving ciphertexts. K was embedded in the code. As encryption starts, the K_is are derived and stored in SRAM. The laser, shown in Figures 7 and 8, is equipped with a YAG laser emitter in three different wavelengths: green, infrared and ultraviolet.

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

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

The chip was decapsulated by chemical etching using a Nisene JetEtch automated acid decapsulator. The decapsulator can be programmed for the chemical opening of different chip types using different ratios of nitric acid (HNO_3) and sulfuric acid (H_2SO_4), at a desired temperature and during a specified time. For opening our chip, we used only nitric acid at 80°C for 40 seconds. The etched chip (Figure 3) successfully passed functional tests before and during fault injection.

As it is very difficult to target the chip's (ALU) (Arithmetic Logic Unit) and inject only a single-byte fault during a very specific instant between the end of `MixColumn` of the 8-th round and before the `MixColumn` of the 9-th round, we decided to target K_8.

Finding the SRAM area containing K_8 and properly tuning the laser's parameters is very time consuming. The number of

[2] 532nm wavelength.

[3] Motorized stepper stage for upright microscopes with 0,1 μm steps.

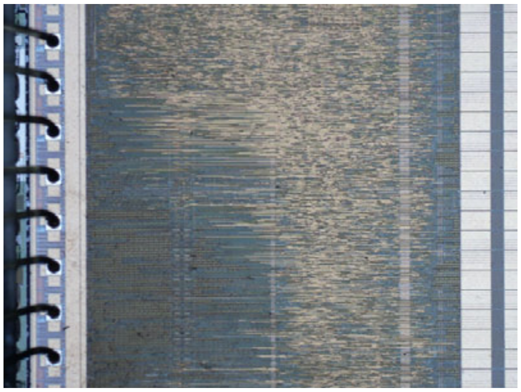

Fig. 4. Decapsulated chip (closeup on SRAM).

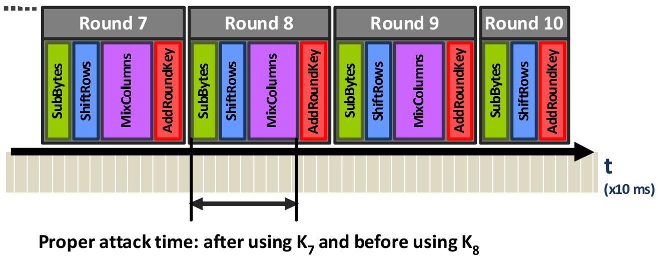

Fig. 5. Attack's timing.

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

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

faults in the ciphertext (C), their position and their contents indicate which round key has been hit. `MixColumns` will amplify any single-byte fault occurring in any K_i preceding K_8 and result in a completely faulty ciphertext (we call such a bad event an *"early fault"*). As shown in Figure 2, a single-byte fault on K_8 changes 4 ciphertext bytes while a fault in K_9 or K_{10} changes only one byte. However, injected faults are not always limited to a single byte and/or to a single K_i. When more than 4 ciphertext bytes are faulty, it is difficult to determine if the observed result is due to an early fault or to several faults in K_8, K_9 and K_{10} (Table I). After finding sensitive SRAM areas that affect the ciphertext, and before targeting K_8, we tried to find the memory cells containing K_{10}. This is necessary for tuning beam-size and energy to limit the fault injection area to very few bytes and preferably one byte. Despite fine-grained energy and spatial control we detected faults in keys neighboring K_8. To overcome this problem, we isolated K_9 and K_{10} from faults and restricted the shot to a 100 μs interval between the use of K_7 and K_8 (Figure 5).

Figure 6 shows how we could confine faults to a single byte

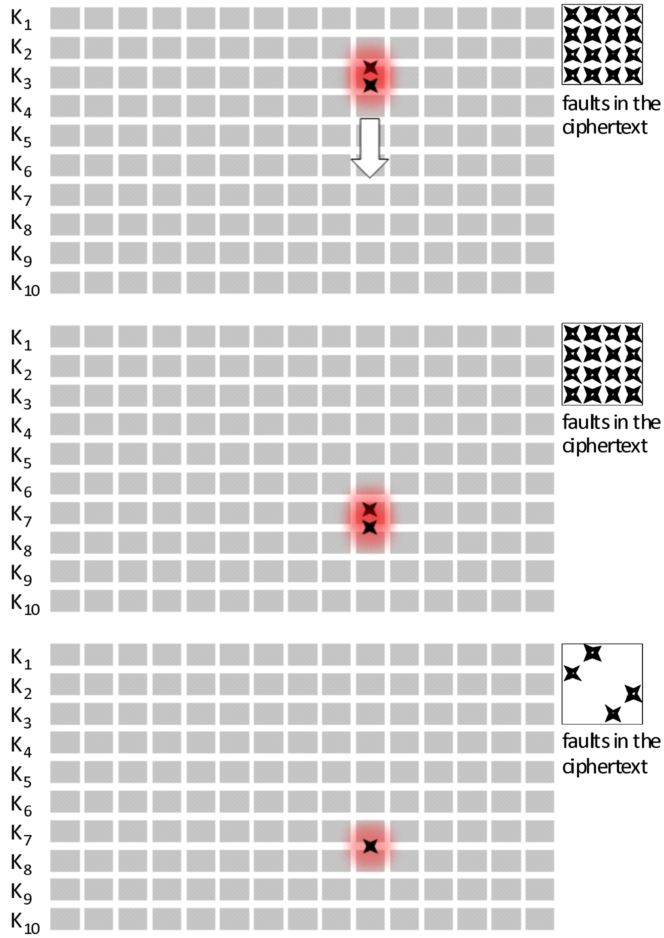

Fig. 6. Exploration process.

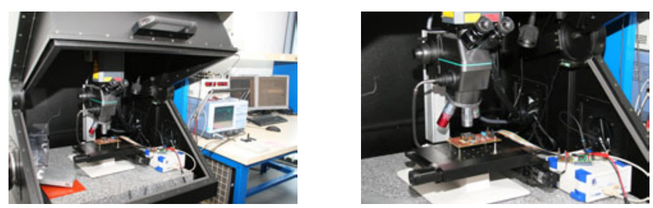

Fig. 7. Laser and target (general overview).

of K_9. When more than one faulty byte existed, we could obtain single byte faults by controlling the laser's shooting time. Figure 6 is just a model of the real SRAM (Figure 4) to describe our technique and does not correspond to real address allocation. We could successfully inject faults into 13 bytes of K_8. This sufficed to run Piret & Quisquater attack.

As shown in the topmost part of Figure 6, we searched K_8's precise storage area by monitoring the number and the type of faults in the ciphertext. Then (middle part of Figure 6), by a precise beam localization, we managed to inject faults only in K_8. This, however, did not turn out to be fully deterministic as sometimes we would also inflict faults to previous round keys. At that point (lowermost part of Figure 6), by fine-tuning spatial and temporal beam localization (just after the use of

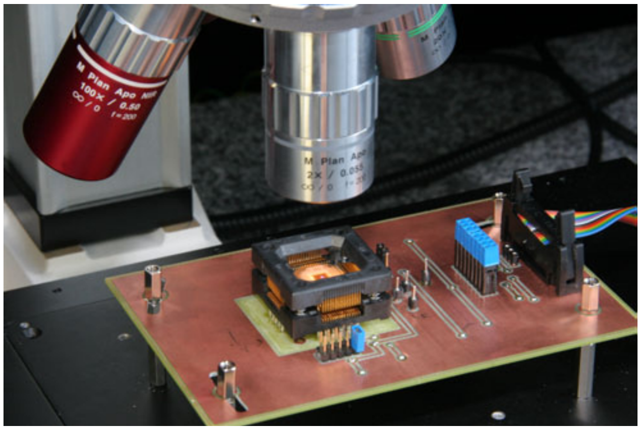

Fig. 8. Laser and target (closeup).

K_7), we managed to restrict the injected faults only to K_8. This is the exact assumption of Piret-Quisquater's scenario.

VI. Conclusion

We implemented Piret-Quisquater's attack [11] using laser fault injection. [11] is usually regarded as the most effective fault attack on AES as it requires only two faulty ciphertexts. This effectiveness comes at the price of stringent fault injection assumptions. The experiments reported in this paper also apply to other attacks (*e.g.* [8], [9]) and underline the possibility to *precisely* modify flags, counters, pointers and other single memory cells that control program flow, in the absence of countermeasures.

In summary, this note's main conclusions are:

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

As we send this paper to press we can already refer the reader to recent *reproducible* single-bit fault injection results [1].

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, In press.

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

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

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

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

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

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

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

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

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

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

[12] 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.

[13] 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