HAL
open science

Single-bit Laser Fault Model in NOR Flash Memories

Alexandre Menu, Jean-Max Dutertre, Jean-Baptiste Rigaud, Brice Colombier, Pierre-Alain Moellic, Jean-Luc Danger

To cite this version:

Alexandre Menu, Jean-Max Dutertre, Jean-Baptiste Rigaud, Brice Colombier, Pierre-Alain Moellic, et al.. Single-bit Laser Fault Model in NOR Flash Memories. 2020 Workshop on Fault Detection and Tolerance in Cryptography (FDTC), Sep 2020, Milan, Italy. pp.41-48, 10.1109/FDTC51366.2020.00013 . hal-03034855

HAL Id: hal-03034855

https://telecom-paris.hal.science/hal-03034855v1

Submitted on 26 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Single-bit Laser Fault Model in NOR Flash Memories: Analysis and Exploitation

Alexandre Menu*, Jean-Max Dutertre*, Jean-Baptiste Rigaud*,
Brice Colombier‡, Pierre-Alain Moëllic†, Jean-Luc Danger§,
*Mines Saint-Etienne, CEA-Tech, Centre CMP, F-13541 Gardanne France
{alexandre.menu, dutertre, rigaud}@emse.fr
†CEA Tech, Centre CMP, Equipe Commune CEA Tech - Mines Saint-Etienne, F-13541 Gardanne France
pierre-alain.moellic@cea.fr
‡Univ Lyon, UJM-Saint-Etienne, CNRS Laboratoire Hubert Curien UMR 5516, F-42023 Saint-Etienne France
b.colombier@univ-st-etienne.fr
§LTCI, Télécom Paris, Institut Mines-télécom, Université Paris Saclay, 75634 Paris Cedex 13 France
jean-luc.danger@telecom-paris.fr

***Abstract*—Laser injection is a powerful fault injection technique with a high spatial accuracy which allows an adversary to efficiently extract the secret information from an electronic device. The control and the repeatability of faults requires the attacker to understand the relation of the fault model to the setup (notably the laser spot size) and the process node of the target device. Most studies on laser fault injection report fault models resulting from a photo-electric current in CMOS transistors. This study provides a black-box analysis of the effect of a photo-electric current in floating-gate transistors of two embedded NOR Flash memories from two different manufacturers. Experimental results demonstrate that single-bit bit-set faults can be injected in code and data without corrupting the Flash memory, even with a laser spot of more than 20 µm in diameter, which is several orders of magnitude larger than the process node of the floating-gate transistors in the experiments. This article also presents the specifics of performing a "safe-error" attack on AES, leveraging the previously detailed single-bit bit-set fault model.**

I. Introduction

Since the late 90s, hardware attacks have been a significant challenge to securing electronic devices from a hardware perspective. Among these attacks, powerful fault injection techniques allow an attacker to alter a device operation in order to extract confidential information or to be granted unauthorized privileges. Fault models are used by the attackers as an abstraction framework on which attack schemes can be based. Designing effective countermeasures against a specific fault model requires the designers to understand the fault model relation to the fault injection technique, device microarchitecture and manufacturing technology.

Local injection techniques are used to disturb specific parts of a microarchitecture without affecting others. Hence, the processing unit, SRAM, Flash memory, and other peripherals of a microcontroller can be individually investigated for faults. The best compromise between cost and locality (both in time and space) is currently achieved with laser fault injection [1].

Among laser fault attacks, fault attacks on Flash memories have drawn less attention than their counterpart on static memory cells. Indeed, computational results are stored in the registers and the SRAM of a microcontroller. Therefore single-bit or single-byte faults injected in these memory locations may be used to corrupt the intermediate results of a computation. A common application of these fault models is found in differential fault attacks [2].

In 2009, Sergei Skorobogatov demonstrates that heating Flash memory cells with a 650 nm wavelength laser source focused down to a 1 µm spot size can be used to erase the value of bits stored in two microcontrollers manufactured in the 0.9 µm technology [3]. While the recovery of a secret key based on this fault model was practical at this time, the implementation of this attack on a recent 90 nm technology is not, as predicted by the author, since the density of Flash memories makes it difficult to erase individual bits [3]. Despite this practical limitation, Obermeier et al. successfully erased protection bits stored in the Flash memory with a selective exposition to UV-C light which did not corrupt the firmware of the device stored in the same memory [4].

The weaknesses of embedded Flash memories with respect to laser fault injection regained attention in 2018,

as Kumar et al. described a transient single-bit bit-reset fault model on the embedded Flash memory of a 8-bit microcontroller [5]. Shortly after, Colombier et al. analysed a similar single-bit bit-set fault model on a 32-bit microcontroller [6]. Colombier et al. discussed the similarities between the two fault models and suggested that the observed faults result from the injection of a parasitic current in the bitlines of the NOR Flash memory. The main implication of this assumption is that the NOR Flash memory organization *constrains* the fault injection process and the resulting fault models.

Other fault models of the Flash memory involve either the current reference of the sense amplifiers [7] or the control logic surrounding the Flash memory [8], [9]. However, the underlying fault mechanism is different from the fault mechanism on floating gate transistors first proposed in [6].

In this article, we provide a black-box analysis of laser-induced faults in two embedded NOR Flash memories from two different manufacturers. Our intent is to validate that the analysis for a given manufacturing process of the fault mechanism reported by [5], [6] is consistent across different manufacturing processes and represents a concrete threat for the microcontrollers that embeds a NOR Flash memory.

Our contributions are the following:

- We propose and describe a fault mechanism for laser-induced faults in NOR Flash memory cells which is consistent with laser fault injection into CMOS logic.
- We analyse bit-set faults and report that injecting 100% repeatable single-bit faults can be achieved with a laser spot which is two orders of magnitude larger than the technology node.
- We exploit this fault model in a practical safe-error attack and successfully extract a 128-bit key with 256 faults.

The rest of this article is organized as follows. Section II introduces the laser fault mechanism observed in our experiments. Section III describes our experimental setup and methodology. Section IV analyses experimental results with respect to the fault mechanism. Section V reports the practical exploitation of this fault model in a safe error attack. Section VI concludes this article.

II. Fault Mechanism

Most microcontrollers embed a non-volatile memory (NVM) to retain code and data when they are not powered. Historically, such memory should have a short read access time and a byte-erase granularity, which has made

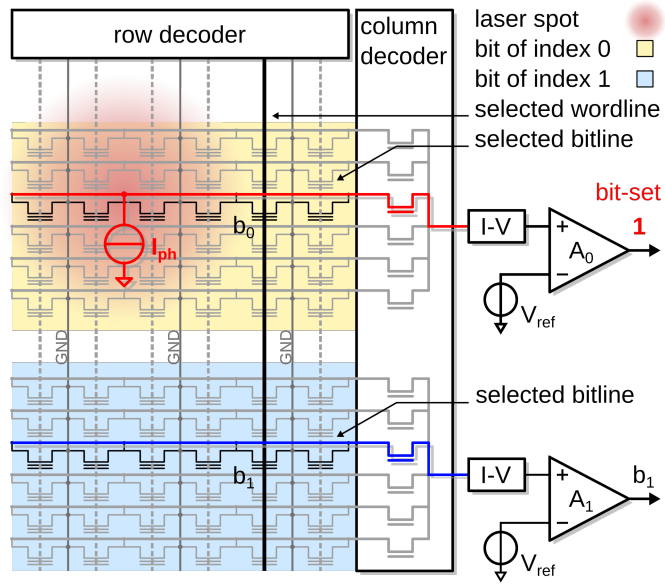

Fig. 1. NOR Flash memory under laser fault attack.

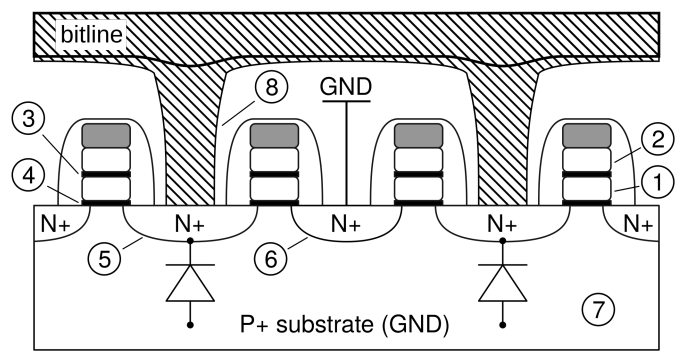

Fig. 2. Schematic cross-section of a NOR Flash column. Floating-gate transistors are made of a polysilicium floating-gate (1) and control gate (2) isolated by an interpoly dielectric (3). The tunnel oxyde (4) isolates the floating-gate from the active areas which are the drain (5) and source (6) of the transistor, and the bulk (7). The drain contacts (8) make the electrical connection between the memory cells and the bitline.

NOR Flash technology the most popular choice in the implementation of embedded NVM for microcontrollers.

In the rest of this article, a *word* refers to a semantic unit of 32 bits of information. In Flash memory, words are stored in an array of floating-gate transistors organized into rows and columns, as depicted in Figure 1. Since each transistor stores a single bit of data, 32 transistors are addressed in parallel to read a 32-bit word.

As a word is being read, its wordline is driven high by the row decoder. Simultaneously, the bitlines of the

columns containing the bits of the word are biased at a fixed potential and connected to the memory sense amplifiers by the selection transistors of the column decoder. As a result, the active columns start sinking current depending on the state of the floating-gate memory cells being selected by the row decoder. Each current is compared to a reference by a sense amplifier and converted into a logic level (`0` or `1`) that is sampled by the output buffer which eventually holds the value of the 32-bit word.

An electrical model of the interaction between laser light and silicon-based electronic circuits was described in [10]. In this model, a photoelectric current is generated in reverse-biased PN junctions of CMOS gates as a result of laser irradiation. This model applies to the drain-bulk junctions of the floating-gate transistors whose columns have been biased at a fixed potential via the column decoder, as illustrated in Figure 2. As a consequence, all the transistors connected to the same bitline may contribute to the parasitic current by collecting charges generated by the laser irradiation in the vicinity of their drain-bulk interface, *even if their wordlines are not selected*.

The photocurrent drawn by the column is injected at one of the input of the sense amplifier, which forces the output of the comparison to either `1` (bit-set fault model) or `0` (bit-reset fault model) depending on the hardware implementation. A high-level schematic of this mechanism is depicted in Figure 1.

Two implications arise from the description of this fault mechanism. First, fault locations should be correlated with the physical placement of the columns in the NOR Flash memory under test. Second, only selected bitlines should be involved in the fault injection process. Thus, the positioning requirements to inject single-bit faults should be relaxed compared to laser injection in static memory cells. In the following sections, we inquire whether these assumptions hold for two embedded NOR Flash memories from different manufacturers.

III. Setup and Methodology

A. Laser injection setup

The laser injection setup used in our experiments consists of a nanosecond near infrared laser source and an optical system which focuses the laser beam inside the silicon substrate of the target under test. The latter is attached to a micrometric XYZ-positioning table, which enables one to scan the whole chip area to disturb specific functionalities of the chip.

The laser source emits a light beam of wavelength 1,064 nm with a maximum power of 3 W, which is able to penetrate several hundreds of micrometers inside the substrate through the backside of the target.

An optical switch shapes the laser pulse intensity to synchronize the disturbance with the target operation. The synchronization is achieved by means of an electrical signal, called a *trigger*, generated by the target under test. The minimal value of the latency from a triggering event to the target irradiation is about 300 ns. The pulse duration can vary from 50 ns to 1 s.

An optical system focuses the laser beam to achieve the lowest possible spot size in the focal plane of the objective lens.

B. Test chips

For the purpose of our experiments, we chose two 32-bit microcontrollers from two different manufacturers. We expected the design and the manufacturing process of the embedded NOR Flash memory to differ between the two manufacturers, since each manufacturer has its own design rules to achieve the lowest possible density. A comparative analysis of faults on two microcontrollers was the first step in generalizing the fault mechanism described in [6].

The first device is a Cortex-M0+ microcontroller implementing the 32-bit Thumb instruction set for ARMv6-M architectures. The device is clocked at 48 MHz. It embeds 256 kB of Flash memory and 32 kB of SRAM. Based on IR imaging, the area of the Flash memory is about 1.29 mm^2, which is about 13% percent of the whole chip area.

The second device is a Cortex-M3 microcontroller implementing the 32-bit Thumb instruction set for ARMv7-M architectures. The device is clocked at 7.37 MHz by an external clock. It embeds 128 kB of Flash memory and 8 kB of SRAM. Based on IR imaging, the area of the Flash memory is about 0.93 mm^2, which is about 10% percent of the whole chip area.

While the bit-set fault model that we describe does *not* depend on the device core architecture, we denote the two microcontrollers Cortex-M0+ and Cortex-M3 in the rest of this article, for the sake of clarity.

C. Test codes highlighting single-bit faults

In a black-box threat model, the assumption is that an attacker has only access to products and documents which are either publicly or commercially available. In this scenario, an attacker can execute arbitrary code

Listing 1 Observation of bit-set faults in a Flash memory transfer with an assembly test routine

```
1  .section .text
2  test_bitset:
3      ldr r0, =.word_in_flash
4      ldr r1, [r0]
5  /* ... */
6      nop
7  /* injection should occur here */
8      ldr r1, [r0]
9      nop
10 /* ... */
11     blx lr
12
13 .section .rodata
14 .word_in_flash:
15 .word 0x00000000
```

sequences on a device from the same family as the target device.

We chose to characterize bit-level faults during memory transfers of *data* triggered by instruction `ldr` and `ldm` of the Thumb instruction set. We report that consistent results were obtained with data and instructions fetched from the two Flash memories.

Bit-set faults were detected with the assembly routine depicted in Listing 1 which loads in register `r1` (line 8) the word `0x00000000` stored in Flash memory (line 15). The value of register `r1` after a laser injection indicates the outputs of the Flash memory which have been set as a result of a laser fault injection. A similar routine which loads the word `0xFFFFFFFF` allowed us to detect bit-reset fault. The conjunction of bit-set faults and bit-reset faults indicates bit-flip faults. However, we report that only bit-set faults were observed inside the two Flash memory arrays, while bit-flip faults where located on one edge of the arrays, as reported in [7]. In the next Section, we characterize the bit-set fault model with respect to the injection parameters.

IV. Fault Model Characterization

Injecting exploitable laser faults requires an attacker to explore at least five dimensions: the injection timing, the pulse duration, the X and Y-coordinates of the laser spot and the power of the laser source. The following section describes the influence of these parameters on the fault model.

Experimental results confirmed the fault mechanism described in Section II for the two microcontrollers. We report in Subsection IV-A that the physical disturbance must be synchronized with the Flash memory operation

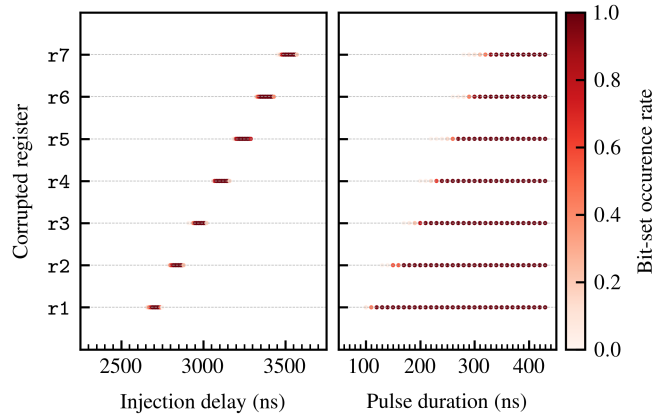

Fig. 3. Bit-set fault injection in bit 18 during the execution of instruction `LDM R0!, {R1-R7}` on the Cortex-M3 microcontroller repeated 100 times with 1 W of power as a function of the delay (left hand side) with a 50 ns pulse duration, and of the pulse duration (right hand side) with a delay of 2,500 ns, by step of 10 ns.

in order to inject a bit-set fault in selected data and instructions fetched from the Flash memory. The position and number of corrupted words can be preciseley selected with the injection timing and the laser pulse duration. We report in Subsection IV-B that the physical disturbance must be localized in the vicinity of selected bitlines. The index of the bits which are set can be precisely selected with the coordinates of the laser spot.

A. Synchronization of the injection on Flash memory accesses

The transfer of a word from Flash memory involves a succession of synchronous operations. According to the fault mechanism described in Section II, a bit is set in a word if a parasitic photocurrent is injected in the corresponding bitline during the current-voltage conversion as the word is fetched from the Flash memory. Thus, the success of the fault injection depends on the synchronization of the disturbance on the target operation.

In the following experiments, we investigated bit-set faults in 32-bit words fetched sequentially from the Flash memory of the Cortex-M0+ and Cortex-M3 microcontrollers. To this end, we injected a fault during the execution of instruction `LDM R0!, {R1-R7}` which loads seven 32-bit words in registers `R1` to `R7` from the address stored in register `R0`.

The rate of single bit-set fault occurences in words fetched from the Flash memory is depicted in Figure 3 and Figure 4 for the Cortex-M3 and the Cortex-M0+ microcontroller respectively as a function of the injection delay and the pulse duration.

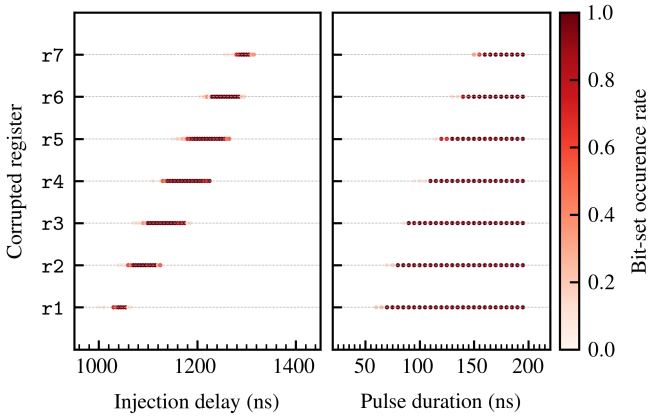

Fig. 4. Bit-set fault injection in bit 7 during the execution of instruction `LDM R0!, {R1-R7}` on the Cortex-M0+ microcontroller repeated 25 times with 1 W of power as a function of the delay (left hand side) with a 50 ns pulse duration, and of the pulse duration (right hand side) with a delay of 1,000 ns, by step of 5 ns.

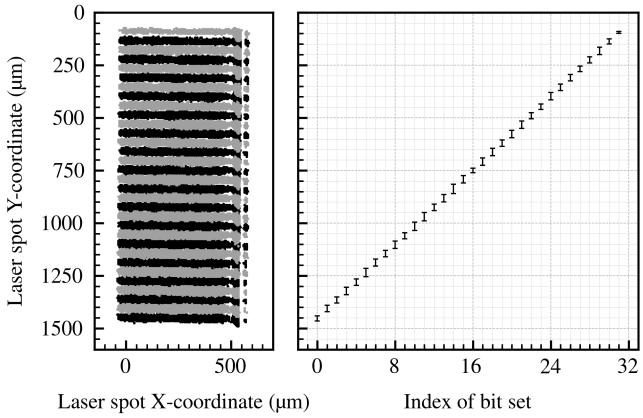

Fig. 5. Location of bit-set faults as a function of the laser spot XY-coordinates by step of 5 µm with 1 W of power and a 20 µm spot size (left hand side) and extremal values of the spot Y-coordinate for which a bit-set fault is observed as a function of the bit index at X-coordinate 250 µm (right hand side) for the Cortex-M3 microcontroller.

The left hand sides of Figure 3 and Figure 4 highlight that each of the seven data transfers can be precisely corrupted with the appropriate injection delay. The time slots in which a bit-set fault could be injected are equally spaced by one (Cortex-M3) or two clock cycles (Cortex-M0+) of the target microcontroller. In the case of the Cortex-M0+ microcontroller, the settings of the NVM controller account for an extra clock cycle (READ WAIT STATE). This repeated patterns match the timing of sequential accesses in Flash memory.

The right hand sides of Figure 3 and Figure 4 show that the number of consecutive data transfers which can be corrupted increases linearly with the duration of the pulse. The seven data transfers of the Cortex-M3 and the Cortex-M0+ can be corrupted with a single 300 ns and 150 ns pulse respectively. Note in this case that the physical effect of the laser shot extends beyond the duration of the pulse. This result is consistant with laser-induced instruction skip of consecutive instructions described in [11]. The spatial distribution of bit-set faults is analyzed in the next paragraph.

B. Spatial distribution of bit-set faults

The Flash memory array is divided into rows of equal size in which consecutive words are stored. Each bit of a word belongs to a column of the memory array, whose corresponding bitline can be selected in order to read the value of the bit (see Section II). The following results demonstrate that the physical locations where bit-set fault could be injected correlate with the physical locations of selected bitlines. Moreover, the size of the rows can be guessed from the fault model dependency on the word address in Flash memory.

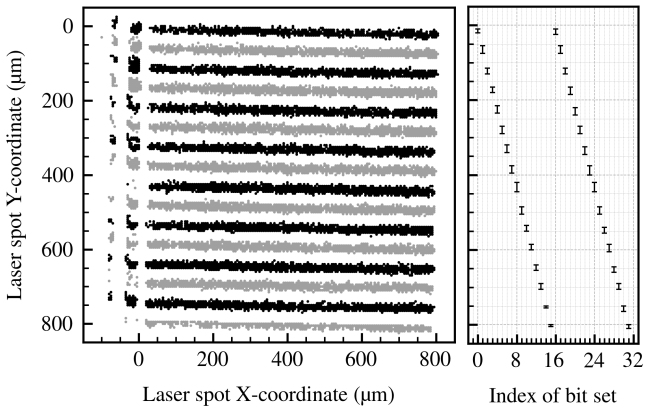

Fig. 6. Location of bit-set faults as a function of the laser spot XY-coordinates by step of 5 µm with 750 mW of power and a a 20 µm spot size (left hand side) and extremal values of the spot Y-coordinate for which a bit-set fault is observed as a function of the bit index at X-coordinate 400 µm (right hand side) for the Cortex-M0+ microcontroller.

A cartography of bit-set faults in the (XY) plane with 1 W of power and a 20 µm spot size is given in Figure 5 for the Cortex-M3 microcontroller. A similar cartography with 750 mW of power and a 20 µm spot size is given in Figure 6 for the Cortex-M0+ microtrontroller.

The two figures exhibit similar rectangular-shaped areas where bit-set fault could be injected. The index of the bit which is set can be selected with the spot Y-coordinate. The influence on the fault model of the spot X-coordinate can be neglected. This observation is con-

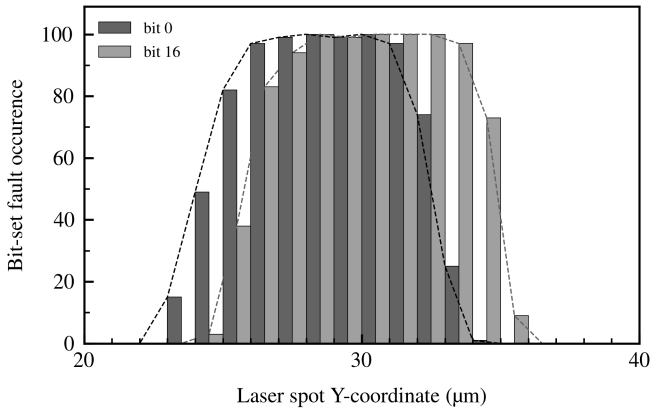

Fig. 7. Distribution of bit-set faults for 100 repetitions of the injection procedure as a function of the laser spot Y-coordinate by step of 1 µm with 400 mW of power and a 5 µm spot size at X-coordinate 300 µm for the cortex-M0+ microcontroller.

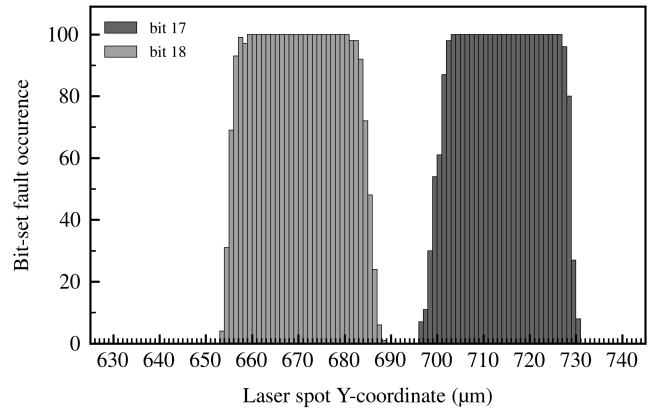

Fig. 8. Distribution of bit-set faults for 100 repetitions of the injection procedure as a function of the laser spot Y-coordinate by step of 1 µm with 1 W of power and a 20 µm spot size at X-coordinate 300 µm for the cortex-M3 microcontroller.

sistent with the fault mechanism described in Section II, in which all the floating gate transistors connected to the same bitline can contribute to the parasitic current.

However, the two different bit mappings in the right hand side of Figure 5 and Figure 6 indicate that the organization of the two memory arrays is different. The bits of equal index modulo 16 are located close to each other in the memory array of the Cortex-M0+ microcontroller, as depicted by the number of bit-set fault occurences as a function of the laser spot Y-coordinate for bit 0 and 16 in Figure 7.

The bell-shaped distribution of bit-set faults along the Y-axis can be explained by the gaussian profile of the laser power density in the (XY) planes, as described in [10]. Although the 1.5 µm distance between the means of the two bell-shaped distributions makes it difficult to inject a single-bit fault in bits of equal index modulo 16, a positionning accuracy of 1 µm makes it possible to injection single-bit fault in either bits with a probability of at least 50%. We demonstrate in Section V that it is sufficient in practice to retrieve the secret key of a cryptosystem stored in the Cortex-M0+ microcontroller.

On the opposite, the 32 bits of the Cortex-M3 microcontroller can be easily distinguished, as depicted in Figure 8.

The width of the area of effect is about 40 µm for bit 17 and bit 18, which is smaller than the spatial periodicity of fault of 45 µm. Single-bit fault were injected with 100% repeatability in the Flash memory of the Cortex-M3 microcontroller with a 20 µm laser spot.

In Figure 9 and Figure 10, the spread of bit-set fault distributions along the Y-axis highlights that the location

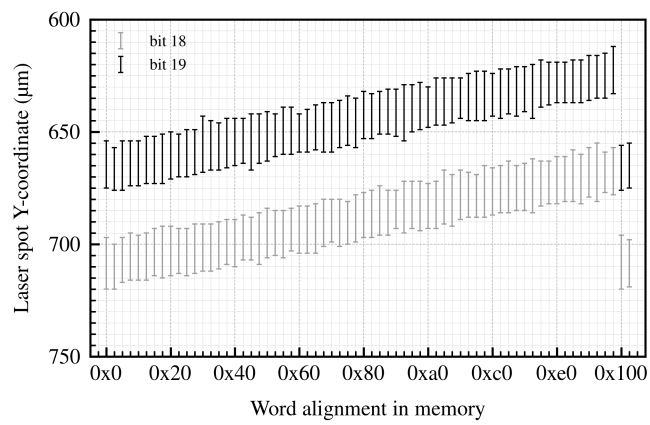

Fig. 9. Spread of bit-set fault distributions along the Y-axis as a function of the offset of the word fetched from the Flash memory with 750 mW of power and a 20 µm spot size at X-coordinate 300 µm for the Cortex-M3 microcontroller.

of selected bitlines changes with the word offset in the Flash memory. The periodicity of the pattern reveals the size of the rows which is 64 words for the Cortex-M3 microcontroller and 16 words for the Cortex-M0+ microcontroller. An estimate of the distance between two adjacent bitline is given by the size of the memory in the Y-direction divided by the number of bitlines. This yields 0.7 µm for the Cortex-M3 microcontroller and 1.5 µm for the Cortex-M0+ microcontroller.

Experimental results indicate that a photocurrent can be injected in the polarized bitlines of a NOR Flash memory array. This new fault model allows an attacker to take advantage of the physical organization of the memory array in order to dynamically set a single bit of his choice for a technology node which is two orders

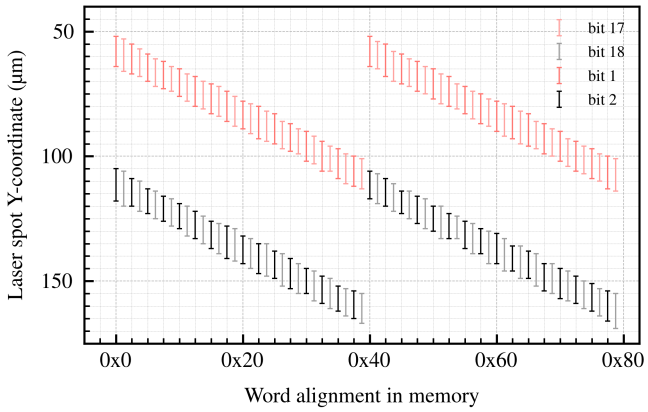

Fig. 10. Spread of bit-set fault distributions along the Y-axis as a function of the offset of the word fetched from the Flash memory with 400 mW of power and a 5 µm spot size at X-coordinate 300 µm for the Cortex-M0+ microcontroller.

of magnitude smaller than the size of the laser spot. In the next section, we leverage this fault model to perform a safe-error attack and retrieve a secret key stored in the Flash memory of the two microcontrollers.

V. Practical Applications

A. Attack principle

Cryptographic algorithms allow two or more parties to securely exchange data by means of a shared secret. This secret is usually a key which has to be stored in the NVM of each electronic deviced involved in the exchange. For most commercial microcontrollers, this memory is the embedded NOR Flash memory, in which code and data are stored. Therefore the fault model described in Section IV can be used to tamper with the transfer of the secret key from the Flash memory, whether the cryptographic algorithm is implemented in software or hardware.

Safe-error attacks are powerful attack schemes which allow an attacker to extract a secret key from a cryptrographic device on the basis of the device response to fault attack [12]. However, the computational complexity of these attack schemes depends on the spatial accuracy of the fault model and a real world scenario requires a bit-level fault model whose practicality has been discussed in [12], [13], [14].

In this article, we report that transient laser-induced faults in the Flash memory of an unprotected microcontroller allow an attacker to retrieve a 128-bit secret key stored in the Flash memory with an unprecedented simplicity. The attack is a direct application of Biham and Shamir attack against the secret key of an unknown cryptosystem [2]. If a bit-set fault is injected in a bit of the secret key whose value is `1`, then the result of a cryptographic computation involving the key should not change. However, if a bit-set fault is injected in a bit whose value is `0`, then a fault should be observed.

Listing 2 Prolog of a generic key scheduling routine in ARM assembly.

```
1  .global AES_128_keyschedule
2  .type   AES_128_keyschedule,%function
3  AES_128_keyschedule:
4      //function prologue
5      push {r4-r11}
6      //load key
7      ldm r0, {r4-r7}
```

B. Attack implementation

The prolog of the key scheduling assembly routine from the AES-128 implementation for Cortex-M3 and Cortex-M4 microcontrollers proposed by Schwabe and Ko [15] is detailed in Listing 2. This routine, whose prototype is `void AES_128_keyschedule(const uint8_t *key, uint8_t *rk)`, takes two 32-bit addresses as inputs: the address of a 128-bit key stored in Flash memory and the address of a buffer in SRAM, where the result of the key schedule is stored. Following the ARM application binary interface, the address of the key is passed in register `r0` and the address of the buffer in register `r1`. The instruction `ldm r0, {r4-r7}` line 7 in Listing 2 loads the 128-bit key in the four 32-bit registers `r4` to `r7`.

We performed a practical recovery of 100 randomly generated 128-bit keys by injecting faults in the assembly routine in Listing 2. The synchronization of the attack on the key scheduling routine is out of the scope of this proof of concept. The reader is refered to [14] for more information about the practical challenges of fault injection synchronization in real world scenarios. For the purpose of the experiment, we synchronized the injection with a trigger signal generated by the target.

Bit-set faults were injected in the Flash memory of the Cortex-M3 microcontroller at 4 different injection times equally spaced by 135 ns in order to fault the four 32-bit words loaded sequentially from the Flash memory. We set the power of the laser source to 1 W, so that the diameter of the laser spot was about the distance between two selected bitlines. The exploration step in the Y-direction was set to 20 µm, which is the size of the area of effect where 100% reproducible faults could be

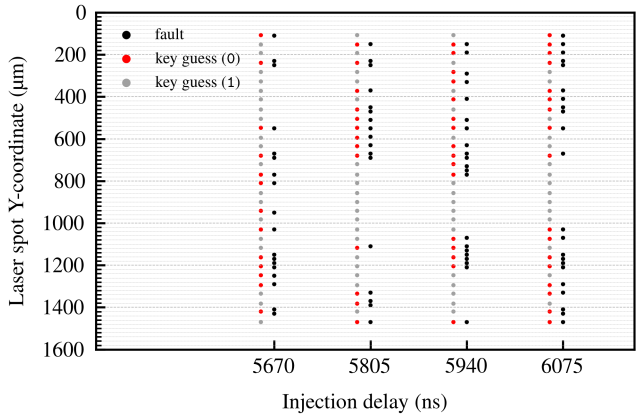

Fig. 11. Safe-error attack on secret key `6FDA6B0D AD03FEF2 9290FC3E 0C5BF924` with 1 W of power at X-coordinate 300 µm for 256 injections. Faults (in black) are matched against reverse-engineered bitline positions (in red and grey) to guess the value of the key.

injected with 1 W of power, as detailed in Section IV. Experimental results for the key `6FDA6B0D AD03FEF2 9290FC3E 0C5BF924` are depicted in Figure 11 We report that 99.5% of 100 randomly generated 128-bit keys could be recovered with this technique.

VI. Conclusion

This paper demonstrates the efficiency of laser induced faults in two NOR Flash memories from different manufacturers. We report that a laser spot of 20 µm in diameter allowed us to inject transient bit-set faults during the read operation of the Flash memories with a bit-level resolution and 100% repeatability. The data stored in the Flash memories remained unaltered. Moreover, data and instructions could be precisely targeted with the appropriate injection timing. As in [11], experimental results highlight the capability to corrupt a large number of consecutive instructions with laser injection. An analysis validated by thorough experiments on two circuits manufactured with two different technologies shows that the single-bit fault model is explained by the physical organization of bitlines which are grouped by bit index. Corrupted bits are either set or reset, depending on the hardware implementation of the sense amplifiers. Moreover, the photo-electric current induced by the laser irradiation has an influence on all memory cells connected to an active bitline, whether the wordlines are selected or not. Hence, the attacker has mainly to know the appropriate injection timing to fault a single bit in a word read from Flash memory, with few constraints on the location of the laser shot. This article demonstrates that powerful attacks as safe-error can be conducted on unprotected microcontrollers with an unprecedented simplicity.

References

[1] J.-M. Dutertre, V. Beroulle, P. Candelier, S. De Castro, L.-B. Faber, M.-L. Flottes, P. Gendrier, D. Hély, R. Leveugle, P. Maistri, G. Di Natale, A. Papadimitriou, and B. Rouzeyre, "Laser fault injection at the CMOS 28 nm technology node: an analysis of the fault model," in *Workshop on Fault Diagnosis and Tolerance in Cryptography*, 2018, pp. 1–6.

[2] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in *Advances in Cryptology — CRYPTO'97*, 1997, pp. 513–525.

[3] S. P. Skorobogatov, "Local heating attacks on flash memory devices," in *International Symposium on Hardware Oriented Security and Trust*, 2009, pp. 1–6.

[4] J. Obermaier and S. Tatschner, "Shedding too much light on a microcontroller's firmware protection," in *Workshop on Offensive Technologies*. USENIX Association, 2017.

[5] D. S. V. Kumar, A. Beckers, J. Balasch, B. Gierlichs, and I. Verbauwhede, "An in-depth and black-box characterization of the effects of laser pulses on atmega328p," in *International Conference on Smart Card Research and Advanced Applications*, vol. 11389, 2018, pp. 156–170.

[6] B. Colombier, A. Menu, J. Dutertre, P. Moëllic, J. Rigaud, and J. Danger, "Laser-induced single-bit faults in flash memory: Instructions corruption on a 32-bit microcontroller," in *International Symposium on Hardware Oriented Security and Trust*, 2019, pp. 1–10.

[7] J. Sakamoto, D. Fujimoto, and T. Matsumoto, "Laser-induced controllable instruction replacement fault attack," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 103-A, no. 1, pp. 11–20, 2020.

[8] S. P. Skorobogatov, "Flash memory bumping attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 6225, 2010, pp. 158–172.

[9] ——, "Optical fault masking attacks," in *Workshop on Fault Diagnosis and Tolerance in Cryptography*, 2010, pp. 23–29.

[10] A. Sarafianos, C. Roscian, J. Dutertre, M. Lisart, and A. Tria, "Electrical modeling of the photoelectric effect induced by a pulsed laser applied to an SRAM cell," *Microelectronics Reliability*, vol. 53, no. 9-11, pp. 1300–1305, 2013.

[11] J. Dutertre, T. Riom, O. Potin, and J. Rigaud, "Experimental analysis of the laser-induced instruction skip fault model," in *Nordic Conference on Secure IT Systems*, vol. 11875, 2019, pp. 221–237.

[12] S. Yen and M. Joye, "Checking before output may not be enough against fault-based cryptanalysis," *IEEE Trans. Computers*, vol. 49, no. 9, pp. 967–970, 2000.

[13] J. Blömer and J. P. Seifert, "Fault based cryptanalysis of the advanced encryption standard (aes)," in *International Conference on Financial Cryptography*, vol. 2742, 2003, pp. 162–181.

[14] P. Loubet-Moundi, D. Vigilant, and F. Olivier, "Static fault attacks on hardware DES registers," *IACR Cryptol. ePrint Arch.*, vol. 2011, p. 531, 2011. [Online]. Available: http://eprint.iacr.org/2011/531

[15] P. Schwabe and K. Stoffelen, "All the AES you need on Cortex-M3 and M4," in *Selected Areas in Cryptography*, vol. 10532, 2016, pp. 180–194.