Laser Profiling for the Back-Side Fault Attacks With a Practical Laser Skip Instruction Attack on AES

Jakub Breier[1], **Dirmanto Jap**[2] **and Chien-Ning Chen**[1]
Physical Analysis and Cryptographic Engineering
Temasek Laboratories@NTU[1]
School of Physical and Mathematical Sciences[2]
Nanyang Technological University, Singapore
{jbreier,chienning}@ntu.edu.sg, dirm0002@e.ntu.edu.sg

Abstract

Laser fault injection is one of the strongest fault injection techniques. It offers a precise area positioning and a precise timing, allowing a high repeatability of experiments.

In our paper we examine possibilities of laser-induced faults that could lead to instruction skips. After the profiling phase we were able to perform an attack on the last *AddRoundKey* operation in AES and to retrieve the secret key with just one faulty and correct ciphertext pair. Our experiments show very high degree of repeatability and 100% success rate with correct laser settings.

Keywords: Laser, Fault Attack, AES, ATmega328P

1 Introduction

Fault attacks on cryptographic devices provide a way to bypass the theoretical security of cryptographic algorithms and also implementations secure to side-channel attacks. Since the first publication, presenting an attack against RSA [2], and a publication aiming at DES [1], many works were presented in this field, proposing mostly theoretical methods on fault attacks on cryptosystems.

Fault injection techniques differ greatly from many points of view – price, repeatability, precision, and user-friendliness. Laser fault injection is considered a very powerful and precise technique, with a high degree of repeatability. Drawback of such a technique is a high price and it requires trained experienced personnel to operate the laser device.

Laser can generate carriers in the silicon substrate which collect in a diffusion area of a target circuit afterwards [7]. The surface of the chip absorbs the energy and creates electron-hole pairs. If this charge is then collected by a diffusion area of a CMOS transistor, it can change the logic output. If this phenomenon occurs in an SRAM or a register, where it flips and locks its state to an opposite one, we call it a Single Event Upset (SEU).

When performing laser fault injections into an integrated circuit, the target area has to be directly accessible. Therefore it is necessary to de-package the chip, usually enclosed in an epoxy package. This can be done either by using specific types of acids (necessary for the front-side de-packaging) or by mechanical grinding and milling the epoxy layers (possible only for the back-side of the chip). Both sides of the chip have different properties that require different laser wavelengths. Since it is not necessary to penetrate the silicon substrate from the front side, it is possible to use the red (808 nm) or the green (532 nm) laser to make the fault injection. However, because of the absorption properties of the silicon [6], we need to use at least near-infrared (1064 nm) laser in order to make the attack possible from the back side of the chip.

For our experiments, we used the 8-bit Atmel ATmega328P 0.35 μm microcontroller, de-packaged from the back side. The main idea of this work is to examine possibilities of an instruction skip induced by a laser. This technique has several advantages over data errors, usually induced in an SRAM or in registers. The power of a laser required for an instruction skip is much lower than the power needed to set/reset bits in memory ($\sim$2% compared to $\sim$35% when using a 20 W near-infrared diode pulse laser), therefore the probability of destroying a chip is lower. The effective area for such attacks is larger, therefore the precise localization takes significantly lesser time than register set/reset attacks. Also, a repeatability of an instruction skip attacks is higher - with correct region and time settings, our experiments on 1000 encryptions show 100% success with 2% laser power.

First, we performed a profiling phase - we loaded the data into 25 different registers of the microcontroller in order to check the precision and the repeatability of the instruction skip attack. We were able to skip loading instructions for each byte separately. Afterwards we performed a simple yet very effective attack on AES last *AddRoundKey* operation, where we were able to skip all the `xor` instructions during one encryption. Therefore, we were able to retrieve the AES-128 secret key with just one faulty and one correct cipher text pair.

The rest of this paper is structured as follows. Section 2 provides an overview of important works in this field. Section 3 describes the setup for our experiments. Next, we provide our profiling method in detail in Section 4, following by practical results on AES in Section 5. Finally, Section 6 summarizes our findings.

2 Related Work

Skorobogatov and Anderson [10] were able to set or reset particular bits of SRAM cells in a microcontroller by using a camera flash and a laser pointer. The microcontroller was decapsulated from the front side and the experiments showed a high vulnerability of CMOS integrated circuits to optical attacks, despite the usage of an inexpensive fault injection equipment.

Dutertre et al. [5] were doing experiments on a 0.35 μm microcontroller. They performed single-byte fault injections in an SRAM and implemented a Piret-Quisquater's fault attack on AES. More detailed results on experiments on SRAM cells are provided in [9]. Authors showed that the *bit-flip* fault model is not feasible with the laser fault injection technique, only the *bit-set/reset* fault model can be achieved. At the same time they performed experiments on ASIC [8], by using a very large laser beam (125x125 μm^2). They could achieve both previously mentioned fault models, exploiting them in order to perform two DFA attacks on AES.

Courbon et al. [3] used a back-side laser fault injection technique to set and reset the state of registers in a 90 nm microcontroller. Later [4], they scanned the surface of a 130 nm microcontroller and identified flip flop patterns in the image. They used such a knowledge to set the area of interest more precisely and therefore it reduced the time needed to perform fault attacks.

3 Setup

In this section we explain the experimental setup which was used for the fault injection:

- **Device Under Test** We have chosen the Atmel ATmega328P microcontroller as the DUT for our experiments. It is an 8-bit microcontroller operating at 16 MHz, manufactured by using a 0.35 μm process. One clock cycle therefore means 62.5 ns from the time point of view. The area of the chip is 3x3 mm^2 large. De-packaged chip is depicted in Figure 1. This chip was mounted on the Arduino UNO[1] board, specifically adjusted for our purposes. The board communicates with the PC using the USBCDC interface.

  All the code for experiments was written in assembly language, by using Arduino programming framework. We set a trigger signal on the board to HIGH (5 V) before performing the operations in order to correctly identify the desired time. The board was mounted on an X-Y positioning table with the step precision 0.05 μm.

- **Laser** We used a near-infrared diode pulse laser with maximal pulse power 20 W. The power was further reduced to 8 W by using a 20x magnifying objective lens. Laser spot size with this lens is 15x3.5 μm^2 and response to trigger pulse is lower than 100 ns according to our experiments.

[1] http://arduino.cc

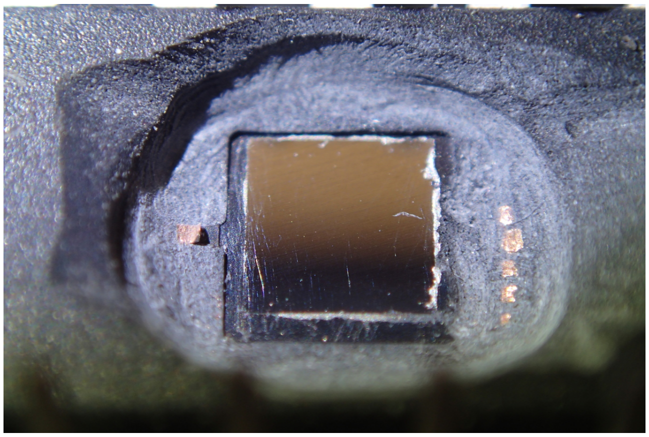

Figure 1: ATmega328P de-packaged from the back side.

4 Laser Fault Injection Profiling

The idea of our attack is to disturb an instruction execution order on the microcontroller and to skip instructions being executed at the moment. The advantage of the laser-induced attacks is in the precise localization of the fault which is hard to be achieved with clock or voltage glitching techniques.

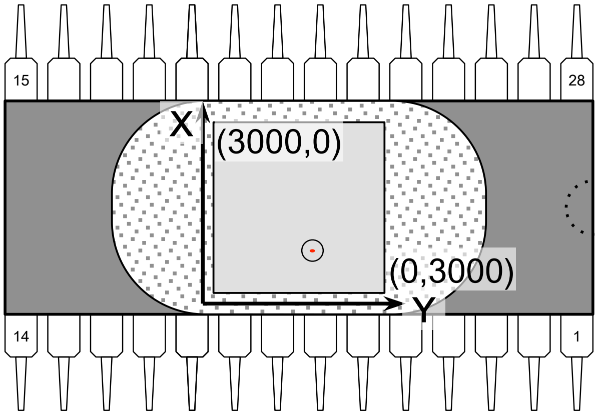

Figure 2: Region of the chip (back side) sensitive to instruction skip fault attack within the whole chip area.

For the profiling phase we used a simple program written in assembly language. It receives 25 bytes from the PC, loads them into registers on the chip, then reads them back and sends them to the PC. Our trigger signal was set immediately before loading the data into registers, so we could precisely target

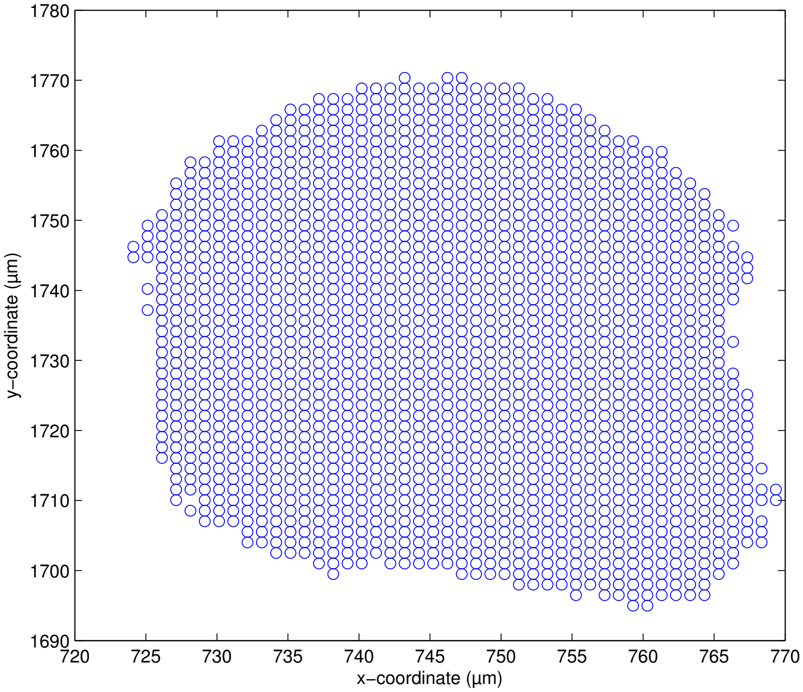

Figure 3: Zoomed region of the chip sensitive to faults.

this phase.

The first important parameter is the position at which it was able to perform the instruction execution disturbance. In Figure 2 we can see this position with respect to the whole area of the chip (back side) and Figure 3 shows the sensitive region more precisely. For the initial localization it is necessary to scan the whole chip. For this preliminary experiment we have set the laser glitch length to 300 ns, step size to 15 μm (200 steps in each direction, resulting to 40,000 experiments in total) and laser power to 1.5%. This localization takes approximately 24 minutes.

When considering instruction skip attacks, it is necessary to set a very precise timing for particular faults. One clock cycle on the microcontroller we used lasts 62.5 ns. Each `load` instruction takes two clock cycles. Following code snippet was repeated 25 times in the program (with different registers):

```
LD    r0,-Y    (2 clock cycles)
EOR   r0,r25   (1 clock cycle)
ST    Y,r0     (2 clock cycles)
```

The `xor` instruction (`EOR`) was used only to simulate the *AddRoundKey* operation of AES. Figure 4 shows different timings for successful attacks on particular bytes. The length of the glitch was always 150 ns, lasting more than two clock cycles, and the power of the laser was set to 1.8%. We can see that it is possible to set the timing very precisely and to skip the desired instructions.

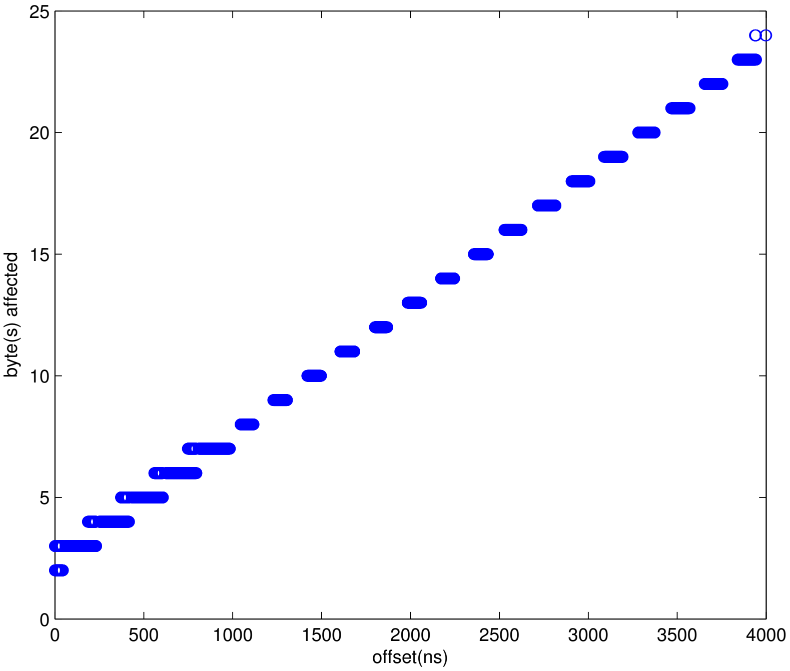

Figure 4: Timing of the instruction skip attack for different bytes.

5 Practical Results on AES

After successful profiling phase we were able to perform a simple yet very powerful attack on AES implementation. The idea of the attack is to skip the `xor` instructions in the last *AddRoundKey*, so that the resulting output of the encryption process is the output of the last *ShiftRows*. Therefore, if we `xor` this output with the correct output, we will get the last round key. With the inverse key schedule it is then easy to get the correct secret key.

Our results showed that with long-enough laser glitch it is possible to skip the whole *AddRoundKey* operation in the last round. Figure 5 shows the number of faulty bytes corresponding to different laser power together with faults that lead to successful key bytes retrieval. We can see that with the laser power around 2% and above it we were able to retrieve all the key bytes with just one fault injected into the encryption process. The Figure also shows us that all the faults injected in this area are instruction skip attacks - all faulty bytes lead to key retrieval. Since the `AddRoundKey` lasts 48 clock cycles (16 `load` and 16 `xor` instructions), the laser glitch length in this case was 3 μs.

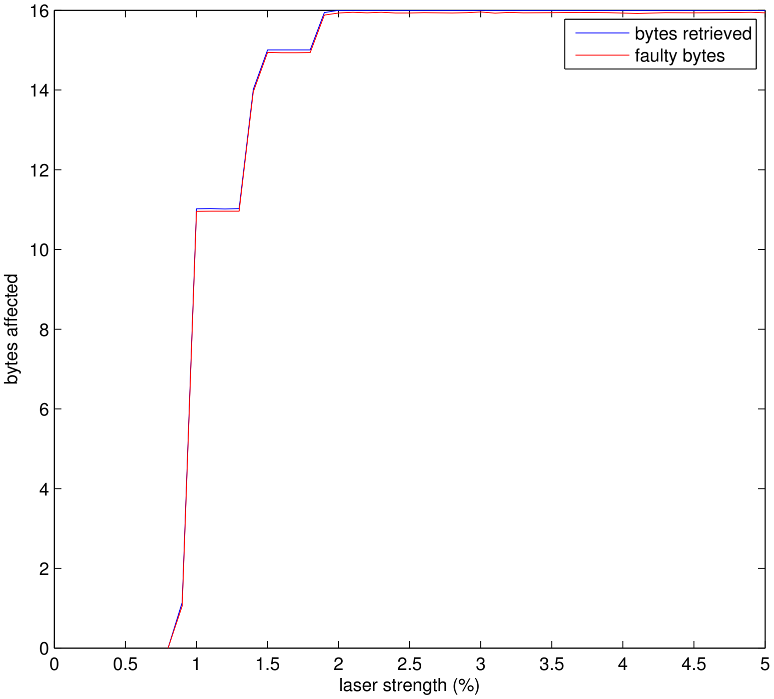

Figure 5: Bytes of the AES output that were faulty together with bytes of the secret key that were retrieved by using the faulty output.

Figure 6 shows the minimal power required to inject the fault into the encryption process. The minimal power which produces any type of fault and affects all 1000 encryptions was 0.68%, however according to previous plot, this power can affect only a few bytes, therefore it is necessary to perform multiple encryptions in order to retrieve the whole key. Note that we used a different random key and plaintext for each encryption.

In Figure 7 we can see the dependency on the position of the laser. In this case all the other parameters were fixed. The area that produces faults in all of 16 bytes is approximately 20x55 μm^2 large ($\sim$0.012% of the whole chip area).

It is worth mentioning that by using this attack model it is easy to break implementations with countermeasures which perform encryption, decryption and then compare plaintexts in order to check for errors. In this case it is necessary to perform a second instruction skip attack during the first AddRoundKey

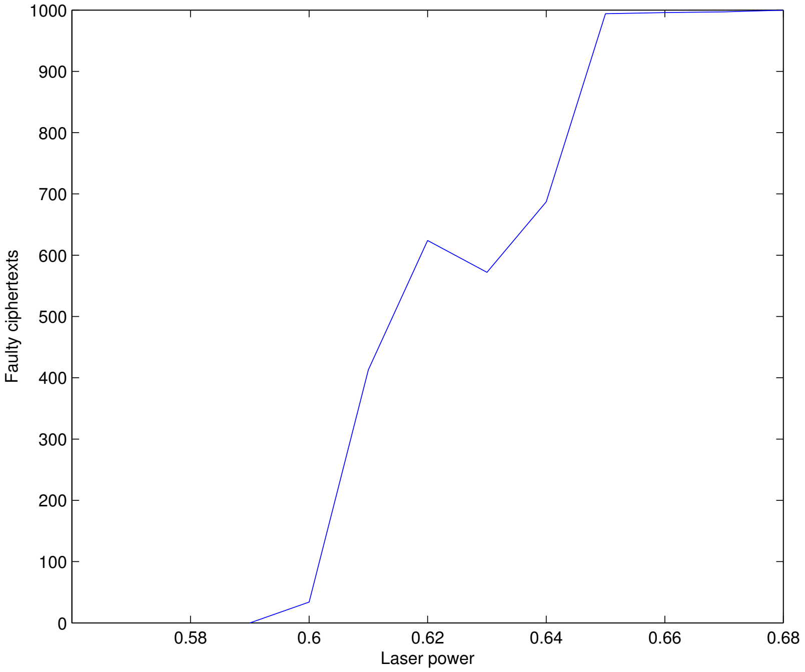

Figure 6: Number of faulty ciphertexts out of 1000 encryptions.

in decryption phase, therefore the resulting plaintext will stay the same as the original input.

Also, some hardware countermeasures can be overcome by this technique, e.g. a light detection sensor on the front side of the chip or an energy sensor which detects higher energy on the chip surface. Since the laser energy required for this type of attack is low, sensor would not be triggered.

On the other hand, the attack can be prevented by using a light detection sensor on the back side of the chip. Loop or instruction counters can make the attack harder but those countermeasures could be overcame as well by precisely skipping counter check.

6 Conclusions

In our work we evaluated a laser-based instruction skip fault attack technique on a microcontroller. Our experiments show a very high repeatability of such attack, together with a high precision of skipping particular instructions in a microcontroller code.

We used a 20 W near-infrared diode pulse laser with 20x magnifying objective

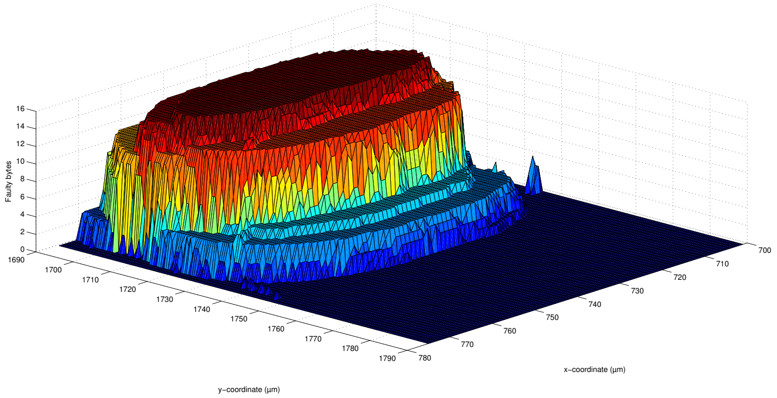

Figure 7: Number of faulty bytes according to position.

to perform an attack on the Atmel ATmega328P 8-bit microcontroller. Laser spot size was 15x3.5 μm^2 and the microcontroller was manufactured by using the 0.35 μm manufacturing process.

After determining correct parameters for instruction skip attacks we evaluated this technique by performing a simple DFA on AES. We were able to skip all the instructions associated with the last *AddRoundKey* operation, resulting to a wrong ciphertext. By `xor`-ing this output with the correct ciphertext we were able to retrieve the last round key and to use the inverse key schedule in order to get the original secret key. This fault attack method is very easy to perform and extremely powerful since it needs only one correct and one faulty ciphertext in order to reveal the full AES key. The success rate was 100% when using 2% laser power and 3 μs glitch length, aiming at the correct region on the chip.

References

[1] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In J. Kaliski, BurtonS., editor, *Advances in Cryptology - CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer Berlin Heidelberg, 1997.

[2] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In *Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'97, pages 37–51, Berlin, Heidelberg, 1997. Springer-Verlag.

[3] F. Courbon, P. Loubet-Moundi, J. Fournier, and A. Tria. Adjusting laser injections for fully controlled faults. In E. Prouff, editor, *Constructive Side-*

Channel Analysis and Secure Design, Lecture Notes in Computer Science, pages 229–242. Springer International Publishing, 2014.

[4] F. Courbon, P. Loubet-Moundi, J. Fournier, and A. Tria. Increasing the efficiency of laser fault injections using fast gate level reverse engineering. In *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*, pages 60–63, May 2014.

[5] J.-M. Dutertre, A.-P. Mirbaha, D. Naccache, and A. Tria. Reproducible single-byte laser fault injection. In *Ph.D. Research in Microelectronics and Electronics (PRIME), 2010 Conference on*, pages 1–4, July 2010.

[6] M. A. Green and M. J. Keevers. Optical properties of intrinsic silicon at 300 k. *Progress in Photovoltaics: Research and Applications*, 3(3):189–192, 1995.

[7] W. Moreno, F. Falquez, and N. Saini. Fault tolerant design validation through laser fault injection [space-based computing systems]. In *Devices, Circuits and Systems, 1998. Proceedings of the 1998 Second IEEE International Caracas Conference on*, pages 132–137, Mar 1998.

[8] C. Roscian, J.-M. Dutertre, and A. Tria. Frontside laser fault injection on cryptosystems - Application to the AES' last round. In *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on*, pages 119–124, June 2013.

[9] C. Roscian, A. Sarafianos, J.-M. Dutertre, and A. Tria. Fault model analysis of laser-induced faults in sram memory cells. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on*, pages 89–98, Aug 2013.

[10] S. Skorobogatov and R. Anderson. Optical Fault Induction Attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 2–12. Springer Berlin Heidelberg, 2003.