Simulations of Optical Emissions for Attacking AES and Masked AES

Guido M. Bertoni[1], Lorenzo Grassi[1,2] and Filippo Melzani[1]

[1] STMicroelectronics, Agrate Brianza (MB), Italy
[2] IAIK, Graz University of Technology, Austria
lorenzo.grassi@iaik.tugraz.at

Abstract. In this paper we present a novel attack based on photonic emission analysis targeting software implementations of AES. We focus on the particular case in which the attacker can collect the photonic emission of a limited number of sense amplifiers (e.g. only one) of the SRAM storing the S-Box. The attack consists in doing hypothesis on the secret key based on the knowledge of the partial output of the SubBytes operation. We also consider the possibility to attack a masked implementation of AES using the photonic emission analysis. In the case of masking, the attacker needs 2 leakages of the same encryption to overcome the randomization of the masks. For our analysis, we assume the same physical setup described in other previous works. Reported results are based on simulations with some hypothesis on the probability of photonic emission of a single transistor.

Keywords:
Photonic side channel, Side channel analysis, Light emission, AES, Boolean Masking, Chosen plaintext attack, Full key recovery.

1 Introduction

Some physical parameters, such as power consumption, electromagnetic radiations, or execution time, depend on processed data and on the performed operations. In the context of cryptographic devices, these data-dependent quantities are called *side-channel leakages*. If the attacker is able to detect vulnerable leakage points and to measure side-channel emanations, she can exploit this dependence to extract information about the secret key. *Side Channel Attacks* (SCA) are the cryptanalytic techniques that consist of analyzing the physical leakage (i.e. measurements of such parameters) produced during the execution of a cryptographic algorithm embedded on a physical device. Example of SCA are Differential/Correlation power analysis (see [5]) and Electro-magnetic analysis.

Protection against these attacks has become a very important and challenging task. In the context of symmetric cryptographic algorithms, the most well-established countermeasure to thwart attacks based on power consumption is masking. The core idea is to mix the sensitive variables with some random values (called *masks*) in order to render every intermediate variable of the computation statistically independent of any sensitive variable. In this way, the measurements of the side-channel leakages are unpredictable to the attacker due to the presence of the masks.

Another possible leakage that can be used to set up a side channel attack is the optical emission. The light emission phenomenon has been mainly studied for failure analysis during the last 25 years and many techniques have been developed to extract and process the light emitted by the electronic components in order to localize different kinds of defects. One of the first uses of photonic emissions in CMOS in a security application was presented in [3], where the authors demonstrate the possibility to set up an attack based on light emitted by the sense amplifiers in order to recover the secret key stored in the microcontroller RAM. In particular, the authors utilize Picosecond Imaging Circuit Analysis (PICA), i.e. one kind of the detector technologies in use today, to spatially recover information about exclusive or operations ($\oplus$) related to the initial AddRoundKey operation of AES. A similar attack has been presented in [9]. In both these works, the authors suppose that an attacker has complete information about the photonic emission, that is she is able to observe the photonic emission of all the sense amplifiers during the reading or/and the writing operations of the SRAM.

Starting from these works, we consider the particular case in which the attacker has only partial information on the photonic emission. The possibility to recover the secret key using only the emissions of a single transistor was already suggested in [6], in [7] and in [2]. In the former paper, the authors perform a Simple Photonic Emission Analysis (SPEA) of a proof-of-concept AES implementation, and they have been able to recover the full AES secret key by monitoring accesses to the S-Box, directly exploiting the side channel leakage of a single transistor of the row inverter. In the second paper (and similarly in the third one), the authors present a Differential Photonic Emission Analysis (DPEA), that is a differential side channel analysis technique applied to the photonic emission measurement of a limited number of sense amplifiers. In particular, they analyze the emission traces of data-dependent regions of the datapath to recover a single bit of the S-Box output and, subsequently, they apply a Difference of Means to recover the full AES secret key. In these previous works, the authors suppose that the S-Box is stored into the SRAM.

In our work, we set up a simple photonic emission attack in the case in which the attacker can observe the photonic emission of a limited number (e.g. only one) of sense amplifiers, that is the photonic emission corresponding to the output of the SubBytes operation. In particular, we focus on the case in which each row of the SRAM stores only one byte (i.e. it is composed of 8 memory cells), which is the same model studied in [3] and [9]. Moreover, in order to minimize the number of plaintexts (and of the tests) that the attacker needs to discover the secret key, we set up a chosen plaintext attack. Finally we consider the possibility to use our Photonic Emission Analysis to attack a software AES implementation protected against first order SCA, even in the previous case of limited knowledge about the photonic emission of the attacker. For our analysis, we have assumed the physical setup described in [6], [7] and [9], and we have focused on the results of these works in order to show our improvements and our new results, which are obtained using a theoretical approach.

The paper is organized as follow. In Sections 2 and 3 we present additional background information on the underlying physics of the photonic emissions in CMOS, the optical emission during the read operation of a SRAM, the AES and the Masked AES algorithm. In section 4 we detail our proposed attack against software implementations of AES-128 in the case of partial information about the photonic emissions, and we set up a chosen plaintext attack. Next, in Section 5 we consider photonic emission analysis on AES with masks as power analysis countermeasure. We conclude in Section 6.

2 Background on Photonic Emission

Currently, most digital circuits are based on CMOS (i.e. Complementary-MOS) technology. CMOS circuits use a combination of complementary and symmetrical pairs of p-type and n-type MOSFETs transistors to implement logic gates and other digital circuits. We restrict photonic emission to CMOS case only.

2.1 Photonic Emissions in CMOS

One of the particularities of CMOS transistors is that photons are emitted during their commutation. Indeed, when a current flows between the source and the drain, the electrons gain energy and accelerate due to the electrical field. At the drain edge of the channel where the field is most intense, this energy is released in radiative transitions, generating photons. The optical emission from a n-channel transistor takes place when the output goes from high to low state, and from a p-channel when it goes from low to high, that is when the transistor opens. This hot-carrier luminescence is dominant in n-type transistors due to the higher mobility of electrons as compared to holes (the photonic emission in a p-type transistor is usually too low to be acquired). Consequently, this phenomenon produces an asymmetric light emission profile that can be used to extract relevant information from the circuit (for more details, see [10] and [12]).

To observe the light emitted, the chip needs to be opened from its backside. The silicon substrate is then mechanically thinned down and polished, in order to decrease the absorption rate of the silicon substrate. The photons emission can be collected by a specific device equipped with a high sensitivity photon sensor mounted on the optical axis of a conventional microscope (see [14] and [13]).

The number of photons emitted by MOS transistors depends on many complex physical aspects, the most important of which are the number of electrons flowing through the MOSFET channel, the probability of each electron to emit a photon and the physical size of the MOSFET. Approximately, the number

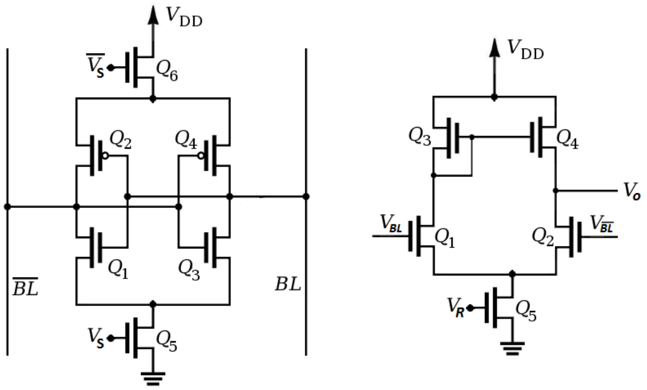

Fig. 1. (a) A sense amplifier with positive feedback - (b) A differential MOS amplifier with a current-mirror load

of emitted photons for each switching transition varies from 10^{-2} to 10^{-4}, but in general only about 5% of the emitted photons reach the detector. Moreover, when they come to the sensor itself, photons are only registered with a certain probability called quantum efficiency (for more details, see [11] and [9]).

Consequently, in contrast to power consumption and electromagnetic field emissions, not every switching of a transistor results in emission of photons. Thus, the absolute number of detectable photons must be integrated over multiple tests.

2.2 Photons emission by the SRAM during the reading operation.

Static random-access memory (SRAM) is a type of semiconductor memory that uses bistable latching circuitry to store each bit. The major part of a memory chip consists of cells in which bits are stored (one bit for each memory cell), and are typically organized in a matrix.

Each cell in the array is connected to one of the 2^M row lines, known as word lines, and to one of the 2^N column lines, known as bit lines. A particular cell is selected for reading or writing by activating its word line, via the row-address decoder, and its bit line, via the column-address decoder. The content of the selected cell is detected by the *sense amplifier*, which provides a full-swing version of it to the data-output terminal of the chip.

During the reading and the writing operations, few photons are emitted both by the memory cell and by the sense amplifier. For both cases, the photonic emission is different (in term of location) if the read bit is a 0-logic or a 1-logic. Thus, knowing the photonic emission during the reading or/and the writing operations, it is possible to discover which bit has been read or/and written. Since a sense amplifier is in general bigger than a memory cell and since the intensity of current flowing through a sense amplifier is greater than that passing through

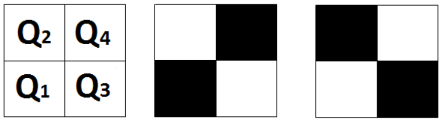

Fig. 2. (a) Schematically representation of a *Sense Amplifier with Positive Feedback* - (b) Photons emission when a 0-logic is read - (c) Photons emission when a 1-logic is read

a memory cell, the number of photons that are emitted by a sense amplifier is greater than those emitted by a memory cell.

Sense amplifiers are essential to the proper operations of SRAMs and a variety of sense-amplifier designs are in use. The two most common models of sense amplifier (shown in Fig. 1) are:

- Sense Amplifier with Positive Feedback;
- Differential MOS Amplifier with a Current-Mirror Load.

In the following, we study the photonic emission of these two models of sense amplifier during the reading operation. Observe that optical emission analysis allows direct observation of the data processed inside semiconductor chips (e.g. data stored in SRAM can be extracted). For more details about the SRAM and the Sense Amplifiers, see [8] (chapter 15).

Sense Amplifier with Positive Feedback. The sense amplifier with positive feedback is a latch formed by cross-coupling two CMOS inverters. Referring to Fig. 2, one inverter is implemented by transistors Q_1 and Q_2, and the other by transistors Q_3 and Q_4. In particular, transistors Q_1 and Q_3 are n-MOS type, while transistors Q_2 and Q_4 are p-MOS type. During the read operation, it can be proven that if the stored bit is a 0-logic, then photons are emitted by transistors Q_2 and Q_3, while they are emitted by transistors Q_1 and Q_4 if the stored bit is a 1-logic (remember that photons are emitted only by MOS in which current flows). Thus, there is a difference in term of location of the photonic emission, but the total number of emitted photons doesn't change.

An example of a real photonic emission described previously can be found in [6], Fig. 3. In this image, you can observe the optical emission of the SRAM cells during the reading operation (remember that the design of a sense amplifier with positive feedback is very similar to that of a memory cell, and that the photonic emission of a memory cell is analogous to that of this kind of sense amplifier in the case of a reading operation). In particular, in this image it is very simple to note the difference (in term of location) of the photonic emission between the case in which the read bit is a 0-logic and the case in which it is a 1-logic.

Numerical model. We want to build a simplified and approximated model that describes the photonic emission of a sense amplifier with positive feedback.

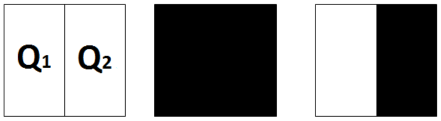

Fig. 3. (a) Schematically representation of a *Differential MOS Amplifier with a Current-Mirror Load* - (b) Photons emission when a 0-logic is read - (c) Photons emission when a 1-logic is read

Let p the following probability:

$$p = Prob(\text{at least one photon is emitted by the transistor during the reading operation of a bit and it is detected by the collector}). \tag{1}$$

Since the number of photons emitted by a p-MOS transistor is negligible compared to the number of photons emitted by a n-MOS transistor, p is well approximated by the probability that at least one photon is emitted by the n-MOS transistor during the reading operation of a bit and that it is detected by the collector. Let us suppose to read the same bit N times and to integrate the photonic emission over the multiple tests, then:

$$Prob(\text{at least one photon is emitted and detected by the collector in } N \text{ reads}) = 1 - (1 - p)^N. \tag{2}$$

Let P_{min} the minimum *chosen* probability that at least one photon is emitted and detected by the collector in N reads. To collect at least one emitted photon in N tests with probability P_{min}, N has to satisfy the following condition:

$$N \geq \frac{\log(1 - P_{min})}{\log(1 - p)}. \tag{3}$$

Differential MOS Amplifier with a Current-Mirror Load. The differential MOS amplifier with a current-mirror load is composed of two identical n-MOS transistors Q_1 and Q_2, as illustrated schematically in Fig. 3. During the reading operation, it can be proven that photons are (mainly) emitted only by the transistor Q_1 if the stored bit is a 1-logic, and that no photons are emitted if the read bit is a 0-logic. Thus, the number of emitted photons depends on which bit has been read.

An example of a real photonic emission described previously can be found in [9], Fig. 7 and 8. In these images, it is very simple to note that there is an optical emission only when the read bit is a 1-logic.

Finally, observe that the photonic emission of a differential MOS amplifier with a current-mirror load can be described by the previous numerical model.

Indeed, for a chosen probability P_{min}, let us suppose as before to read the same bit N times (where N is defined in (3)), and to integrate the photonic emission over the multiple tests. Then, the read bit is a 1-logic if at least one photon is emitted in N reads, otherwise it is a 0-logic with probability P_{min}.

3 Background on AES

The Advanced Encryption Standard (AES) is a secret key encryption algorithm based on the Rijndael cipher [1]. AES can process data blocks of 128 bits, using cipher keys with lengths of 128, 192 and 256 bits, and operates on a 4×4 matrix of bytes, named the state. The algorithm is specified as a number of identical rounds (except for the last one) that transform the input plaintext into the ciphertext. AES consists of 10, 12 and 14 rounds for 128-, 192- and 256-bit keys, respectively.

Since our attack exploits the leakage obtained during the beginning of the first round of AES, we present only the two beginning operations that are executed until then, namely *AddRoundKey* and *SubBytes*. In the AddRoundKey step, each byte of the plaintext is combined with the corresponding byte of the secret key, using the exclusive or operation ($\oplus$). In the SubBytes step, each byte of the state is replaced with another according to a fixed 8-bit lookup table, denoted S-Box. The used S-Box is constructed by combining the multiplicative inverse function over $GF(2^8)$ (known to have good non-linearity properties) with an invertible affine transformation. This operation provides the non-linearity in the algorithm.

3.1 The Masked AES Algorithm

The core idea of masking is to conceal all intermediate values with some random values called *masks*, in order to make the leakage measurements unpredictable. For every execution of the algorithm, new masks are generated. Hence, the attacker does not know the masks. The masks are added at the (very) beginning of the algorithm to the plaintext. During the execution of the algorithm, one needs to take care that every intermediate value stays masked. Obviously, a correct masking scheme doesn't have to modify the ciphering.

For our work we decided to focus on the first order masking AES proposed by *C. Herbst et al.* in [4]. We only present the masking scheme of AddRoundKey and SubBytes operations of the beginning of the first round of AES.

In this scheme, we use two (byte) masks, M and M', as the input and the output masks for the masked SubBytes operation. At the start of each AES encryption, we pre-compute a masked SubBytes table S-Box$'$ such that $\forall x \in GF(2^8)$

$$\text{S-Box}'(x \oplus M) = \text{S-Box}(x) \oplus M'. \tag{4}$$

At the beginning of the first round, the plaintext byte p is masked with M (i.e. $p_M = p \oplus M$), and then the AddRoundKey operation is performed on p_M. Then, the SubBytes operation with the table S-Box$'$ is performed and this changes the mask to M' (indeed: S-Box$'(p_M \oplus k) =$ S-Box$(p \oplus k) \oplus M')$.

4 Photonic side channel attacks on AES

The typical strategy of side channel attacks is to reveal each byte of the key separately. Thus, for the following we work on a fixed but arbitrary single byte of the key, of the plaintext and of the intermediate state.

If an attacker is able to know the photons emission of all the sense amplifiers of the SRAM, she can use this knowledge to find the key in a very simple way. In particular, she can discover the secret key using the photons that are emitted by the sense amplifiers during the reading of the secret key from the SRAM, for instance when needed for the AddRoundKey operation (see [3] and [9] for a detailed exposition).

Let us suppose now that an attacker is only able to collect the photons that are emitted by a limited number of sense amplifiers. In this case, if the attacker can know at least 6 bits for each byte of the secret key (that is 96 bits of the complete key) using for example the previous method, then she can simply discover the remaining 32 bits (and so the complete secret key) using a brute force attack. Otherwise, in general the attacker is not able to discover the complete secret key using only the knowledge of the photons that are emitted by less than 6 sense amplifiers during its reading.

To discover the secret key in this case we concentrate on the output of the SubBytes operation, and in particular on the photons that are emitted by (some) sense amplifiers during the reading of the output of the SubBytes operation. Indeed, note that the knowledge of one bit of the output of the S-Box allows the attacker to do some hypothesis on the input of the S-Box (and so on the byte of the secret key), because each bit of the output of the S-Box depends on all the bits of its input.

We emphasize that the possibility to recover a single bit of the S-Box output by analyzing emission traces of data-dependent regions of the datapath has been proven in [7]. More generally, the possibility to recover a bit by analyzing the photonic emission and using the techniques described in subsection 2.2 has already been proven in practice in [6] and [9].

4.1 Monitoring the SRAM

We consider the case in which the S-Box is contained within the SRAM, which led us to consider possible side channels that exist within this memory. As in [6], our attack needs an initial spatial analysis to allow for at least a basic understanding of the chip's functionality and the organization of the SRAM to identify the S-Box within memory. We refer to [6] for a detailed explanation of the initial spatial analysis of the SRAM.

We start showing our attack in the simple case in which each row of the SRAM is composed by 8 memory cells, i.e. each row of the SRAM stores one byte (observe that this is the same model studied in [3] and [9]). Then we will generalize the models considered for the attack.

In the simple model, we suppose that there is an area of the SRAM where each row stores one byte of the S-Box and where all the r-th bits of each byte of the S-Box are on the r-th bit line. That is, during the read operation, the r-th bit of the output of the SubBytes operation is read and amplified by the r-th sense amplifier. Moreover, we suppose that the attacker can observe only the sense amplifier of the single (fixed) column r of the SRAM, i.e. she is only able to collect the photons that are emitted by the r-th sense amplifier. Thus, using this photonic emission, the attacker is able to discover the r-th bit of the output of the SubBytes operation. In the next subsection, we describe how she can use this knowledge to do hypothesis on the secret key.

4.2 Key recovery in the simple model

Let us suppose that an attacker discovers that the r-th bit of the output of the S-Box for an input message m is b. Using this information she can eliminate all the candidates $k \in GF(2^8)$ of the secret key byte such that

$$\text{S-Box}(m \oplus k)_r \neq b, \tag{5}$$

where $\text{S-Box}(x)_r$ denotes the r-th bit of the output of $\text{S-Box}(x)$.

The idea is to repeat this simple operation with different plaintexts m until the attacker recovers the byte k of the secret key.

Let m_1 the first plaintext used by the attacker, and let b_1 the r-th bit of the output of the SubBytes operation of the exclusive or of m_1 and of the secret key byte. We define K_1 as the set of all possible candidates of the secret key byte after the first step:

$$K_1 = \{k \in GF(2^8) \,|\, \text{S-Box}(m_1 \oplus k)_r = b_1\}, \tag{6}$$

It is simple to verify that $|K_1| = \frac{1}{2}|GF(2^8)| = 128$ for each choice of m_1 (remember that the S-Box is a bijective function), where $|K|$ denotes the cardinality of the set K.

If the attacker iterates this procedure using different plaintexts, she can discover the secret key. Indeed, let us suppose to be at the $(h-1)$-th step (where $h \geq 2$) and let K_{h-1} the set of all the possible candidates of the key byte at this step (where $|K_{h-1}| > 1$). As previously, using the h-th plaintext byte m_h (where $m_h \neq m_1, ..., m_{h-1}$), she can eliminate other candidates of the key byte. Thus, starting from K_{h-1}, let K_h defined as:

$$K_h = \{k \in K_{h-1} \,|\, \text{S-Box}(m_h \oplus k)_r = b_h\}, \tag{7}$$

where b_h is defined as before. Observe that $|K_h| \leq |K_{h-1}|$. If $|K_h| = 1$, then the attacker has found the secret key byte, otherwise she has to repeat this procedure for a new plaintext byte m_{h+1}.

The attacker surely discovers the byte of the secret key using a finite number of different plaintext bytes. Indeed, we have verified by computer simulations that for each $k_1, k_2 \in GF(2^8)$ such that $k_1 \neq k_2$ and for each $r \in \{1, ..., 8\}$, there exists at least one $m \in GF(2^8)$ such that

$$\text{S-Box}(k_1 \oplus m)_r \neq \text{S-Box}(k_2 \oplus m)_r.$$

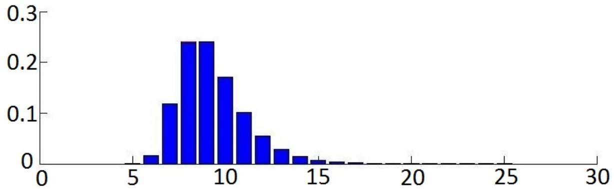

Fig. 4. The histogram shows (on the vertical axis) the probability that an attacker needs a certain number of plaintexts (on the horizontal axis) to recover the byte of the secret key. The histogram was obtained with 250 000 simulations.

This implies that for each sequence $m_1, m_2, ..., m_{256}$, there exists an integer h such that $2 \leq h \leq 256$ and $|K_h| = 1$.

The number of plaintexts that an attacker needs to discover the byte of the secret key is not fixed if the plaintexts are chosen in a random way. In particular, using computer simulations, we found that if she chooses the plaintexts in a random way, then:

- the average number of plaintexts she needs to recover the secret key is about 9.3;
- in the best case, she needs only 5 (different) plaintexts to recover the secret key;
- in the worst case, she needs up to 146 (different) plaintexts to recover the secret key.

The probability that an attacker needs a certain number of plaintexts to recover the byte of the secret key is showed in the histogram in Fig. 4.

To explain the fact that the number of plaintexts is not constant if they are chosen in a random way, consider the following example. Let the secret key byte $k = 0\text{x}65$, the first plaintext byte $m_1 = 0\text{x}27$ and $r = 6$. The number of key candidates after the first step is 128. Let m_2 the second plaintext byte. Then, the number of possible keys after the second step depends on the choice of m_2:

- if $m_2 = 0\text{x}2B$, the number of key candidates after the second step is 72;
- if $m_2 = 0\text{x}10$, the number of key candidates after the second step is 64;
- if $m_2 = 0\text{x}C5$, the number of key candidates after the second step is 60.

This situation also occurs in the next steps and this is the reason why the number of plaintexts that the attacker needs is not constant.

4.3 Chosen plaintext attack in the simple model

If the attacker has the possibility to do a chosen plaintext attack, she can choose the plaintexts $m_1, m_2, ...$ in order to minimize the number of plaintexts that she needs to recover the byte of the secret key. In the following, we show a way to choose the plaintexts such that the attacker needs only 8 different plaintexts to recover the secret key. Moreover, the following algorithm (to choose the plaintexts) can be easily generalized to more generic models.

The first plaintext byte m_1 can be chosen in a random way, because, as we have seen, any choice of m_1 halves the number of the candidates of the secret key.

The h-th plaintext byte m_h ($h \in \{2, ..., 8\}$) has to satisfy the following condition[1]:

$$\begin{aligned} &|\{k \in K_{h-1} \,|\, \text{S-Box}(m_h \oplus k)_r = 0\}| = \\ &= |\{k \in K_{h-1} \,|\, \text{S-Box}(m_h \oplus k)_r = 1\}|. \end{aligned} \tag{8}$$

Observe that this condition implies that $m_h \neq m_1, ..., m_{h-1}$. If m_h satisfies the condition (8) and if K_h is defined as in (7), it is simple to verify that

$$|K_h| = \frac{1}{2}|K_{h-1}| = \frac{1}{2^h}|GF(2^8)| = \frac{256}{2^h}.$$

Thus $|K_8| = 1$, that is K_8 contains only the secret key.

If there is no m_h that satisfies (8), the idea is to choose m_h that minimizes the following quantity:

$$\begin{aligned} abs(&|\{k \in K_{h-1} \,|\, \text{S-Box}(m_h \oplus k)_r = 0\}| - \\ &|\{k \in K_{h-1} \,|\, \text{S-Box}(m_h \oplus k)_r = 1\}|). \end{aligned} \tag{9}$$

In this case, the number of plaintexts that the attacker needs to find the byte of the secret key can be greater than 8, but using this method she can still minimize the number of plaintexts.

Why does m_h have to satisfy the condition (8)? We define:

$$A = \{k \in K_{h-1} \,|\, \text{S-Box}(m_h \oplus \tilde{k})_r = 0\},$$

$$B = \{k \in K_{h-1} \,|\, \text{S-Box}(m_h \oplus \tilde{k})_r = 1\},$$

where $\tilde{k}$ is the secret key. Observe that $|A| + |B| = |K_{h-1}|$. It is simple to prove that if $|A| > \frac{1}{2}|K_{h-1}|$ (or $|A| < \frac{1}{2}|K_{h-1}|$), the number of the key candidates after the h-th step is greater than $\frac{1}{2}|K_{h-1}|$ with probability 0.5 (or it is less than $\frac{1}{2}|K_{h-1}|$ with probability 0.5, respectively). Instead if $|A| = |B| = \frac{1}{2}|K_{h-1}|$, then the number of the key candidates after the h-th step is equal to $\frac{1}{2}|K_{h-1}|$ with probability 1.

We repeated the previous computer simulations using the method described above. In all these tests the attacker always needs 8 plaintexts to find the secret key. From computer simulations, we can say that:

[1] The condition (8) is equivalent to the following condition:

$$\sum_{k \in K_{h-1}} \text{S-Box}(m_h \oplus k)_r = \frac{|K_{h-1}|}{2}.$$

- the random choice is better in 13.3% of cases;
- the two methods are equivalent in 24.0% of cases;
- the above method is better in 60.7% of cases.

It is simple to note that the plaintext bytes $m_1, ..., m_8$ can be precomputed for each possible output bit sequence $b_1, ..., b_8$.

Another way to choose the second plaintext $\boldsymbol{m_2}$. Given K_1 and m_1, an equivalent condition that the second plaintext m_2 has to satisfy is:

$$\begin{aligned} &|\{k \in K_1 \,|\, k \oplus m_1 \oplus m_2 \in K_1\}| = \\ &= |\{k \in K_1 \,|\, k \oplus m_1 \oplus m_2 \notin K_1\}|. \end{aligned} \tag{10}$$

It is very important to observe that this condition works only for the choice of the second plaintext m_2. Additionally one can observe that the condition (10) is independent from the S-Box functionality.

To prove this condition, we introduce two sets A and B:

$$A = \{k \in GF(2^8) \,|\, \text{S-Box}(m_1 \oplus k)_r = j\}$$

$$B = \{k \in GF(2^8) \,|\, \text{S-Box}(m_2 \oplus k)_r = l\},$$

where $j, l \in \{0, 1\}$. Using:

- $|A \cap B| + |A \cap B^C| = |A|$,
- if $j = l$, then

$$\begin{aligned} |A \cap B| &= |\{k \in A \,|\, k \oplus m_1 \oplus m_2 \in A\}| \\ |A \cap B^C| &= |\{k \in A \,|\, k \oplus m_1 \oplus m_2 \notin A\}|, \end{aligned}$$

- if $j \neq l$, then

$$\begin{aligned} |A \cap B^C| &= |\{k \in A \,|\, k \oplus m_1 \oplus m_2 \in A\}| \\ |A \cap B| &= |\{k \in A \,|\, k \oplus m_1 \oplus m_2 \notin A\}|, \end{aligned}$$

it is simple to prove the condition (10).

4.4 Key recovery in the generic model

The method described in the previous subsections can be extended to more generic models. In particular, if the attacker can observe S ($1 \leq S \leq 8$) sense amplifiers, our method changes very little (only in the definition of (7)) and the number of plaintexts/tests that the attacker needs to recover the secret key decreases.

More interesting is the case in which the number of sense amplifiers is greater than 8 (that is they are 2^N with $N > 3$). In this case, the idea is to repeat our attack in the same way. Anyway, it works efficiently only when the attacker can observe at least one of every 8 sense amplifiers. We plan to further investigate more specifically the attack for this case in a forthcoming work.

In both the previous cases, it is easy to generalize and to adapt the chosen plaintext attack described in the previous subsection to these generic models.

Finally, if the attacker can observe both the photonic emission of the row decoder and of the sense amplifiers, she can combine our attack with the one described in [6].

5 Photonic side channel attacks on masked AES

As we said before, the common approach to secure implementations of symmetric cryptographic algorithms against power analysis attacks is randomize the key-dependent data by the addition of one or several random *masks*. Our goal is to understand if AES with power analysis countermeasure can be considered secure against photonic side channel attacks. In particular, for our work we consider the efficient first order masking AES proposed in [4] and explained in subsection 3.1.

As previously, we focus on the case in which each row of the SRAM stores one byte (that is each row is composed of 8 memory cells) and we suppose that the masks are stored into the SRAM. However, the following analysis holds for more generic models.

5.1 Key recovery

Let us suppose for the moment that an attacker can observe all the sense amplifiers of the SRAM, which means that she is able to collect the photons that are emitted by all the sense amplifiers. In this case, the masking scheme for the AES is completely useless against photonic emission analysis. Indeed, as in the case of unmasked AES, the attacker can discover the secret key using the photons that are emitted by the sense amplifiers during the reading of the key from the SRAM (required for the AddRoundKey operation). Since the read secret key is always the same, she can repeat this operation as many time as she wants, in order to integrate the photonic emission over multiple tests (remember that the number of detectable photons is so low that it needs to be averaged over multiple tests). Using this procedure, she can obtain the secret key in the same way as the unmasked AES.

Consider now the case in which an attacker can observe only a limited number (e.g. one) of sense amplifiers of the SRAM, that is she is only able to know the photons emission of a limited number (e.g. one) of sense amplifiers. As previously, a possible way to discover all bits of each byte of the secret key is to attack the output of the SubBytes operation. However, using this method there is an important difference between the masked and the unmasked case that must be taken into account. In the case of unmasked AES, an attacker can repeat the encryption as many time as she wants, and she can integrate the photonic emissions over multiple tests in order to recover the read bit. Instead, in the case of masked AES, the attacker can not do this, because the intermediate values (and so the photonic emissions) are different for every encryption due to the presence of the masks. Thus, if an attacker is not able to understand if the read bit is 0- or 1-logic with only one photonic emission, she can not attack masked AES using the output of the SubBytes operation in this particular case. For the following, we assume that it is sufficient one photonic emission to understand if the read bit is 0- or 1-logic. Observe that this assumption is (at the moment) unrealistic (for example it means that there is no noise), but it is the best situation for the attacker.

With this assumption, the attacker must use two leakages to attack the masked AES, due to the presence of the masks. It is very important to note that these two leakages must be of the same encryption, that is the masks of the two leakages have to be the same. There are several possibilities about the leakages that can be used to implement the attack. We consider the two following cases:

- two different bytes of the masked message (with the same masks);
- one byte of the masked message and of the associated mask.

Another interesting possibility is to attack the key schedule to recover the secret key (remember that each round key depends on the initial secret key): we plan to further investigate this possibility in a forthcoming work.

5.2 Two different bytes of the masked message (with the same masks).

Let us suppose that an attacker knows the r-th bit of the i-th and of the j-th byte of the output of the masked SubBytes operation ($i \neq j$). We denote respectively by b_i and b_j these two bits, and by m_i and m_j the i-th and the j-th byte of the plaintext. As before, the idea is to use this information to eliminate some key candidates. The procedure is very similar to that explained in section 4, but in this case we attack two different bytes of the secret key simultaneously.

Let K_1 the set of all the possible candidates of the i-th and of the j-th byte of the secret key after the first step[2]:

$$\begin{aligned} K_1 = \{(k_i, k_j) \in GF(2^8) \times GF(2^8) \,|\, &\text{S-Box}(m_i \oplus k_i)_r \\ &\oplus \text{S-Box}(m_j \oplus k_j)_r = b_i \oplus b_j\}. \end{aligned} \tag{11}$$

Observe that for each $x, y \in GF(2^8)$:

$$\text{S-Box}'(x)_r \oplus \text{S-Box}'(y)_r = \text{S-Box}(x)_r \oplus \text{S-Box}(y)_r.$$

Using different couples of plaintext bytes m_i and m_j, the attacker can eliminate other candidates of the key repeating the above procedure, until she finds the

[2] In this subsection, we omit the index (h) of the step on m and on b for an easier reading.

secret key. We define K_h as the set of all possible candidates of the key after the h-th step:

$$K_h = \{(k_i, k_j) \in K_{h-1} \,|\, \text{S-Box}(m_i \oplus k_i)_r \oplus \text{S-Box}(m_j \oplus k_j)_r = b_i \oplus b_j\}. \quad (12)$$

Also in this case, if the attacker chooses the plaintext in a random way, the number of plaintexts that she needs to discover the secret key is not constant. At the h-th step, if she has the possibility to do a chosen plaintext attack, the chosen plaintext bytes m_i and m_j have to satisfy the following condition:

$$\begin{aligned} &|\{(k_i, k_j) \in K_{h-1} \,|\, \text{S-Box}(m_i \oplus k_i)_r \oplus \text{S-Box}(m_j \oplus k_j)_r = 0\}| = \\ &= |\{(k_i, k_j) \in K_{h-1} \,|\, \text{S-Box}(m_i \oplus k_i)_r \oplus \text{S-Box}(m_j \oplus k_j)_r = 1\}|, \end{aligned}$$

in order to minimize the number of plaintexts that the attacker needs to recover the bytes of the secret key. It is simple to prove that if m_i and m_j satisfy the previous condition, then $|K_h| = \frac{1}{2}|K_{h-1}|$.

5.3 One byte of the masked message and of the associated mask.

Let us suppose that an attacker knows the r-th bit of the output of the masked SubBytes operation for a plaintext byte m (denoted b) and the r-th bit of the masked M' (denoted $M'^{(r)}$). Also in this case, she can use these information to eliminate some candidates of the key and to discover the byte of the secret key. In this particular case, the attack is completely equivalent to that described in section 4. For this reason, we refer to that section for a complete explanation of the attack, and we limit ourselves to re-define the set K_h used in (7) and in (8).

We define K_1 as the set of all the possible candidates of the secret key byte after the first step:

$$K_1 = \{k \in GF(2^8) \,|\, \text{S-Box}(m_1 \oplus k)_r = b_1 \oplus M_1'^{(r)}\}, \quad (13)$$

and, in the same way, let K_h the set of all the possible candidates of the secret key byte after the h-th step:

$$K_h = \{k \in K_{h-1} \,|\, \text{S-Box}(m_h \oplus k)_r = b_h \oplus M_h'^{(r)}\}, \quad (14)$$

where, as before, $m_h \neq m_1, ..., m_{h-1}$. Remember that:
$\text{S-Box}'(x)_r = \text{S-Box}(x)_r \oplus M'^{(r)}$ for each $x \in GF(2^8)$.

If the attacker has the possibility to do a chosen plaintext attack, she can choose the plaintexts using the algorithm (8) described in subsection 4.3, in order to minimize the number of plaintexts/tests.

Observe that, during the encryption, the mask M' could be read several times depending on how masked AES is implemented. For example, during the pre-computation of the masked S-Box′, the mask can be read 256 times, i.e. one for each input/output of the S-Box, or it can be read only 1 time and then stored in a working register. If the mask M' is read more times, then the attacker may have more opportunities to have two photons emissions (one for the mask and one for the plaintext) in the same encryption.

Table 1. The following table gives an estimate of the number of tests that the attacker needs to do in order to discover the key for different values of p, P_{min} and R. Remember that these numbers are obtained with simple and approximated models.

p	P_{min}	(unmasked) AES	Masked AES & $R = 256$	Masked AES & $R = 1$
10^{-4}	95 %	29 960	1 170 210	299 573 230
10^{-4}	99.99 %	92 100	3 597 785	921 034 050
10^{-5}	95 %	299 575	117 020 795	29 957 322 740
10^{-5}	99.99 %	921 050	359 778 930	92 103 403 750

5.4 Numerical model and comparison

We want to compare the number of acquisitions required by an attacker in order to discover one or more bits of the output of the SubBytes operation, both in the unmasked and in the masked AES case. In this second case, we consider only the case in which the attacker uses the two leakages of one byte of the masked message and of the associated mask M': remember that the two leakages have to be of the same encryption (i.e. the masks of the two leakages must be the same). Moreover, in both cases we suppose that the knowledge of at least 1 emitted photon is sufficient for the attacker to discover which bit has been read.

In the unmasked AES case, the required number of tests for each plaintext is given by the equation (3). In a similar way, it can be proven that the required number of encryptions/tests for each plaintext in the masked AES case is given by

$$N \geq \frac{\log(1 - P_{min})}{\log(1 - R \cdot p^2)}, \tag{15}$$

where p is defined in (1), P_{min} is the chosen probability that at least one photon is emitted and detected by the collector in N encryptions, and R is the number of times that the mask M' is read during the encryption process. Observe that the probability that there is at least one photonic emission in R reads of the same bit of the mask M' is given by $1 - (1 - p)^R$, but since $0 < p \ll 1$, then $1 - (1 - p)^R \simeq 1 - (1 - R \cdot p) = R \cdot p$. The quantity p^2 in (15) depends on the fact that the attacker needs at least two photonic emissions (respectively, at least one for the bit $M'^{(r)}$ of the mask and at least one for the bit S-Box$'(p_M \oplus k)_r$) for the same encryption.

Table 1 gives an estimation of the minimum number of tests that an attacker needs to do in order to discover the secret key for some different values of p, P_{min} and R. We emphasize that these numbers are obtained with simple and approximated models, and they are useful only in order to do a simple comparison between the unmasked and masked case.

The relationship between the number of tests in the masked and in the unmasked AES case is given by:

$$\frac{N_{\text{masked AES}}}{N_{\text{(unmasked) AES}}} = \frac{\log(1-p)}{\log(1-R \cdot p^2)} \simeq \frac{1}{R \cdot p} > 1. \tag{16}$$

If $(R \cdot p)^{-1} \gg 1$, then the number of tests in the masked case is much bigger than in the unmasked case. In this case, the time that the attacker needs to collect the two leakages in the same encryption can be so long that the attack can become unworkable.

6 Conclusion

In this work we have presented a novel attack based on photonic emission analysis against software implementations of AES-128. We have mainly analyzed the case in which the attacker can collect the photonic emission of a limited number (e.g. only one) of sense amplifiers and in which each row of the SRAM stores only one byte. Based on the state of the art and on the capability of the real equipment, the analysis of a single spot is shown to be a realistic scenario. The presented attack can easily be adopted to AES-192 and AES-256.

Attacking masked AES is another novel result reported in this paper. In this case the attacker needs 2 leakages of the same encryption to overcome the randomization of the masks. Moreover, the number of acquisitions needed by the attacker increases by a factor proportional to p^{-1} with respect to the unmasked AES case, where p is the probability that at least one photon is emitted by the transistor and detected by the collector during the read operation. Since p is practically very low and since it is not possible to integrate the photonic emission over multiple tests, a simple photonic emission analysis seems to be not practical to attack masked AES.

Acknowledegments. The work has been supported in part by the Austrian Science Fund (project P26494-N15).

References

1. J. Daemen and V. Rijmen, *The design of Rijndael: AES - the Advanced Encryption Standard.* Springer Verlag, 2002.
2. J. Di-Battista, J. C. Courrege, B. Rouzeyre, L. Torres, and P. Perdu, “When Failure Analysis Meets Side-Channel Attacks,” in *Cryptographic Hardware and Embedded Systems, CHES 2010*, ser. Lecture Notes in Computer Science, S. Mangard and F.-X. Standaert, Eds. Springer Berlin Heidelberg, 2010, vol. 6225, pp. 188–202.
3. J. Ferrigno and M. Hlavàĉ, “When AES blinks: introducing optical side channel,” *Information Security, IET*, vol. 2, no. 3, pp. 94–98, 2008.
4. C. Herbst, E. Oswald, and S. Mangard, “An AES Smart Card Implementation Resistant to Power Analysis Attacks,” in *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings.* Springer, 2006, pp. 239–252.

5. P. C. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, “Introduction to differential power analysis,” *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011.
6. A. Schlösser, D. Nedospasov, J. Krämer, S. Orlic, and J.-P. Seifert, “Simple Photonic Emission Analysis of AES,” in *Cryptographic Hardware and Embedded Systems CHES 2012*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7428, pp. 41–57.
7. ——, “Differential Photonic Emission Analysis,” in *Constructive Side-Channel Analysis and Secure Design*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 7864, pp. 1–16.
8. A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 6th ed. Oxford University Press, 2009.
9. S. P. Skorobogatov, “Using Optical Emission Analysis for Estimating Contribution to Power Analysis,” in *6th Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE Computer Society, 2009, pp. 111–119.
10. F. Stellari, F. Zappa, S. Cova, and L. Vendrame, “Tools for non-invasive optical characterization of CMOS circuits,” in *Electron Devices Meeting, 1999. IEDM ’99. Technical Digest. International*, Dec 1999, pp. 487–490.
11. F. Stellari, F. Zappa, M. Ghioni, and S. Cova, “Non-Invasive Optical Characterisation Technique for Fast Switching CMOS Circuits,” in *Solid-State Device Research Conference, 1999. Proceeding of the 29th European*, vol. 1, Sept 1999, pp. 172–175.
12. A. Tosi, F. Stellari, F. Zappa, and S. Cova, “Hot-carrier luminescence: comparison of different CMOS technologies,” in *European Solid-State Device Research, 2003. ESSDERC ’03. 33rd Conference on*, Sept 2003, pp. 351–354.
13. J. Tsang and M. Fischetti, “Why hot carrier emission based timing probes will work for 50 nm, 1V CMOS technologies,” *Microelectronics Reliability*, vol. 41, no. 9 - 10, pp. 1465 – 1470, 2001.
14. S. Villa, A. L. Lacaita, and A. Pacelli, “Photon emission from hot electrons in silicon,” *Phys. Rev. B*, vol. 52, pp. 10 993–10 999, 1995.