Differential Photonic Emission Analysis

Juliane Krämer[1], Dmitry Nedospasov[1],
Alexander Schlösser[2], and Jean-Pierre Seifert[1]

[1] Security in Telecommunications, Technische Universität Berlin, Germany
{juliane,dmitry,jpseifert}@sec.t-labs.tu-berlin.de
[2] Optical Technologies, Technische Universität Berlin, Germany
schloesser@opttech.tu-berlin.de

Abstract. This work presents the first differential side channel analysis to exploit photonic emissions. We call this form of analysis Differential Photonic Emission Analysis (DPEA). After identifying a suitable area for the analysis, our system captures photonic emissions from switching transistors and relates them to the program running in the chip. The subsequent differential analysis reveals the secret key. We recovered leakage from the datapath's driving inverters of a proof of concept AES-128 implementation. We successfully performed DPEA and were able to recover the full AES secret key from the photonic emissions. The system costs for an attack are comparable to power analysis techniques and the presented approach allows for AES key recovery in a relevant amount of time. Thus, this work extends the research on the photonic side channel and emphasizes that the photonic side channel poses a serious threat to modern secure ICs.

Keywords: Photonic side channel, differential analysis, AES, full key recovery.

1 Introduction

Side channel attacks are a significant research area since the seminal papers of Kocher in 1996 and 1999, which introduced the timing [12] and the power side channel [13]. Since then, other side channels, e.g., electromagnetic (EM) radiation [10,18], and applications, e.g., cache timing attacks [4], and various analysis methods, such as template attacks [5,6] and mutual information analysis [3], have been developed.

Most side channel attacks focus on system-wide information leakage, whereas the photonic side channel, which was first introduced in 2008 [9], also allows selective in-depth analysis of specific parts of the hardware. Since attacks targeting single transistors are possible with Photonic Emission Analysis (PEA), the selectivity of photonic emission analysis greatly exceeds the selectivity of EMA. Targeting specific elements of an integrated circuit (IC) results in significantly better signal-to-noise ratios and potentially, signals can be captured that consist entirely of leakage. However, due to the huge cost and complexity of the

E. Prouff (Ed.): COSADE 2013, LNCS 7864, pp. 1–16, 2013.

necessary equipment used in [9], the photonic side channel was not regarded as a realistic threat at that time.

Since then, new research has introduced new applications and has even demonstrated that PEA can be realized with low-cost equipment. It was exploited for reverse engineering [17] and for attacking the DES algorithm on an FPGA [8]. Simple Photonic Emission Analysis (SPEA) was recently introduced [19]. It was shown that SPEA is a powerful tool, and an attack on AES was demonstrated. However, this concrete attack can be made significantly harder by randomization or on-the-fly calculation of the `SubBytes` operation (which come at the cost of expensive calculations).

This work extends the state of the art of PEA by introducing the Differential Photonic Emission Analysis (DPEA) and presenting the first successful differential analysis on measurements of photonic emissions leading to the revelation of the AES-128 secret key. Since the attack presented in this paper is both low-cost and efficient, this work demonstrates that photonic side channel attacks pose a serious threat to modern secure ICs.

The main contributions of this paper are as follows:

A novel methodology: Differential Photonic Emission Analysis. We introduce Differential Photonic Emission Analysis (DPEA), which is the adaptation of differential analysis methods to measurements of photonic emissions, analogous to Differential Power Analysis (DPA, [13]) and Differential Electromagnetic Analysis (DEMA, [2,18]).

Results of a successful DPEA of AES. Using DPEA in combination with a low-cost optoelectronic system, we were able to correctly recover the full secret key of a proof of concept (PoC) AES-128 implementation running on a common microcontroller, the ATmega328P. The process technology of the ATmega328P is approximately 350 nm. We exploited the photonic leakage of an SRAM buffer, monitoring the first `SubBytes` operation.

Organization. The rest of this work is structured as follows: In Section 2 we present additional background information on photonic emissions in CMOS, the AES algorithm and related work. In Section 3, we introduce Differential Photonic Emission Analysis (DPEA). We explain the optoelectronic setup used in this work and details of both the hardware and software of our PoC AES implementation in Section 4. In Section 5, we explain the DPEA that we successfully conducted against AES-128. Finally, we conclude in Section 6.

2 Background

2.1 Photonic Emissions in CMOS

CMOS transistors emit near-infrared light, so-called Hot-Carrier Luminescence, when current flows through the conductive channel. This is due to parasitic

radiative transitions, which the accelerated electrons undergo at the drain edge of the channel [22]. As a result of the increased mobility of electrons compared to holes, this effect is dominant in n-type transistors. For a standard CMOS-inverter this creates data-dependent photonic emissions in the following way: If the input is changed from 0 to 1 the n-type transistor will carry a current, emitting photons. For the inverse case the p-type transistor will carry the current, emitting less photons.

These emissions pose a side channel comparable to power consumption and electromagnetic field emissions. However, in contrast to those, photonic emission is a statistical process and measurements result in discrete count numbers. In addition, the absolute number of detectable photons is very low and needs to be averaged over many switching operations. To maximize detection efficiency, modern ICs are best observed from the backside as interconnect layers obstruct observation from the frontside. Detection efficiency, specifically for silicon detector technologies, can be further boosted by mechanically thinning the IC substrate, as described in Section 4.2.

2.2 The AES Algorithm

The Advanced Encryption Standard (AES) is a secret key encryption algorithm based on the Rijndael cipher [7]. AES has a fixed block size of 128 input bits and operates on a 4×4 matrix of bytes, named the state. Depending on the length of the key, which is 128, 192, or 256 bits, the cipher is termed AES-128, AES-192, or AES-256. The algorithm is specified as a number of rounds that transform the input plaintext into the ciphertext. AES consists of 10, 12 and 14 rounds for 128-, 192- and 256-bit keys, respectively. Each round consists of 4 different operations (`SubBytes`, `ShiftRows`, `MixColumns`, `AddRoundKey`), except for the final round, which skips the `MixColumns` operation. Additionally, there is an `AddRoundKey` operation before the first round. Regarding AES-128, the secret 128-bit key is used for this initial `AddRoundKey` operation, whereas for the 10 rounds, each 128-bit round key is derived from the original secret key using Rijndael's key schedule.

Since our attack exploits the leakage obtained during the beginning of the first round of AES, we present only the two operations that are executed until then, namely `AddRoundKey` and `SubBytes`. In the `AddRoundKey` step, each byte of the state is combined with a byte of the round key using the exclusive or operation ($\bigoplus$). In the `SubBytes` step, each byte of the state is replaced with its corresponding entry in a fixed 8-bit lookup table referred to as the S-Box. This is the only operation that provides non-linearity in the algorithm. Instead of using this lookup table, the substitution value can also be calculated on the fly. However, due to costly inverse calculations in $\mathbf{GF}(2^8)$ otherwise, precomputed tables are used most often. In contrast to the implementation-specific Simple Photonic Emission Analysis [19], for the attack presented in this paper the implementation of the `SubBytes` operation is irrelevant.

2.3 Related Work

The first use of photonic emissions in CMOS for a side channel attack was presented in [9], where the authors utilize Picosecond Imaging Circuit Analysis (PICA), one of the most complex detector technologies in use today, to spatially recover information about exclusive or operations ($\oplus$) related to the initial `AddRoundKey` operation of AES. More recently, an integrated PICA system and laser stimulation techniques were used to attack a DES implementation on an FPGA [8]. The authors showed that the optical side channel can be used for differential analysis and partiallly recovered the secret key using temporally resolved measurements. However, the use of equipment worth more than two million Euros does not make such analysis particularly relevant. Additionally, the analysis strongly relied on a specific fixed state of the transistors before each measurement. This was achieved by alternating between relevant plaintexts and zero messages. Full key recovery was not presented. Most recently, a novel low-cost optoelectronic setup for time- and spatially resolved analysis of photonic emissions was presented [19]. The authors also introduced a corresponding methodology, named Simple Photonic Emission Analysis (SPEA). They successfully performed SPEA of a proof of concept AES implementation and were able to recover the full AES secret key by monitoring accesses to the S-Box. In the field of electromagnetic side channel analysis, location-dependent leakage was successfully exploited in an attack on an elliptic curve scalar multiplication implementation on an FPGA using a near-field EM probe [11]. The authors demonstrated that location-dependent leakage can be used in a template attack and countermeasures against system-wide leakage can thus be circumvented.

In [20], photonic emissions were used for basic reverse engineering. Low-cost equipment was used to capture photonic emissions via backside analysis and gain basic information about the operations executed on an IC. Recently, a novel, automated methodology for performing functional analysis of integrated circuits was introduced [17]. By selectively executing code on a given chip, the resulting optical emission images yield critical information about the chip's functional layout. This methodology provides an efficient way to isolate potential points of interest and can also serve as a basis for DPEA.

3 Differential Photonic Emission Analysis

Definition 1. *Differential Photonic Emission Analysis (DPEA) reveals the secret key of a cryptographic device based on a large number of traces of photonic emissions that have been recorded while the device encrypts or decrypts different data. The data dependency of the intensity of the photonic emissions at certain points in time, which do not have to be known in advance, is exploited by a statistical analysis.*

In case this analysis does not reveal the whole secret key but leaves only so many key candidates that a brute force attack gets feasible, we also call such analysis a

DPEA. In contrast to Simple Photonic Emission Analysis, DPEA attacks require a more complex analysis, since a visual inspection of the traces will not be sufficient. In contrast to Differential Power Analysis, detailed knowledge about the cryptographic device might be necessary, or at least advantageous. We show in Sections 4 and 5, how the detailed knowledge about the Device Under Test (DUT) allows for the spatial identification of potential points of interest and thus, for more efficient attacks.

We use the following agreement and notation throughout the remainder of this paper: The attacker collects D traces $\mathbf{t}_d$ of photonic emissions, $d \in \mathcal{D} = \{0, \ldots, D-1\}$. The trace $\mathbf{t}_d$ is recorded while the device encrypts or decrypts the data block d with the use of the fixed secret key $\tilde{k}$, which originates from the K-element set of all possible keys $\mathcal{K} = \{0, \ldots, K-1\}$. Each trace consists of N points in time, i.e., N is the length of the traces and thus, $\mathbf{t}_d = (t_{d,1}, \ldots, t_{d,N})$. The traces $\mathbf{t}_d$ and their components $t_{d,i}, i \in \mathcal{I} = \{1, \ldots, N\}$, respectively, thus refer to real photonic emissions and each $t_{d,i}$, corresponds to a number of count events, cf. Section 2.1. In addition, a DPEA also requires a function $h : \mathcal{D} \times \mathcal{K} \times \mathcal{I} \to Y$, which describes potential photonic emissions, based on data $d \in \mathcal{D}$, key hypothesis $k \in \mathcal{K}$, and point in time $i \in \mathcal{I}$. It maps to a discrete image set Y. We call the function h a hypothesis function, since it models hypothetical emission values based on an assumption about the relation between the cryptographic operation running on the DUT and the photonic emissions. The hypothesis function may, or may not, depend on a given point in time. In case the point in time does not have to be considered, we just write $h(d, k)$ and omit the third argument. The hypothesis function may map into the set $Y = \{0, 1\}$, as well as into other sets, e.g., the 9-element set $Y = \{0, 1, \ldots, 8\}$. The latter could be used in case the attacked algorithm operates on bytes, i.e., $\mathcal{D} = \mathcal{K} = \{0, 1, \ldots, 255\}$, and the hypothesis function uses the Hamming weight (HW) or Hamming distance (HD) model.

Each byte $b \in \{0, 1, \ldots, 255\}$ is of the binary form $b = b_7|b_6|b_5|b_4|b_3|b_2|b_1|b_0$ with $b_i \in \{0, 1\}\ \forall\ i \in \{0, \ldots, 7\}$, i.e., we count the bits starting with the least significant bit. Thus, for any $x \in \mathbb{N}$ in any notation, x_2 denotes the respective bit 2 of x. A DPEA reveals the key $\tilde{k}$ of the attacked device by interrelating the traces $\mathbf{t}_d$ and hypotheses $h(d, k, i)\ \forall\ d \in \mathcal{D}, k \in \mathcal{K}$. This is done based on statistical analyses. These can be as simple as a correlation coefficient, but also be considerably more complex, as is the case for DPA, e.g., [3,5,14].

4 Experimental Setup

4.1 Optoelectronic System

The experimental setup used in this work was identical to the one employed in [19]. A silicon-based CCD and an InGaAs-based single avalanche photo diode (APD) serve as primary detectors and are connected to the device under test via a custom-built near-infrared microscope and an FPGA-based controller. The Si-CCD captures photons below 1μm wavelength and is used to provide

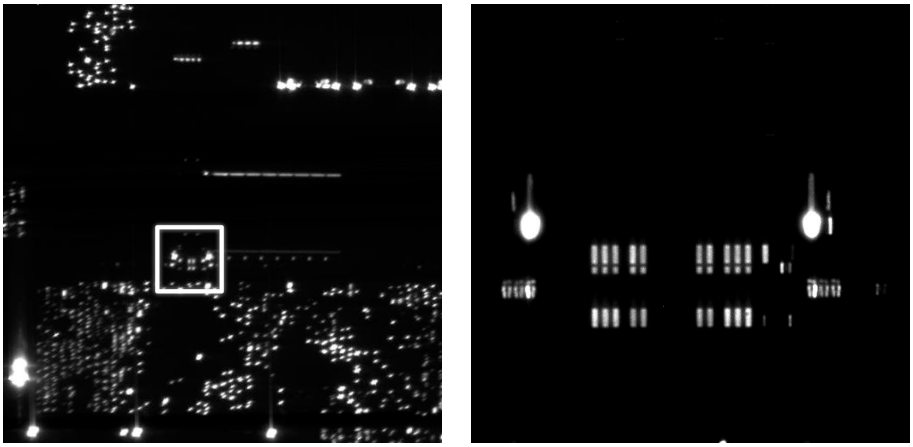
spatial orientation by creating emission images of the DUT. The acquisition time necessary for adequate emission images ranges from a few seconds to many minutes. It depends strongly on the supply voltage, the switching frequency of the transistors and the substrate thickness. The InGaAs-APD is used to perform time-resolved measurements of specific points of interest on the DUT, as identified by emission images. It detects photons above 1μm wavelength and therefore does not require substrate thinning. The APD is operated in gated Geiger mode to alleviate technology-inherent noise. This means that, in contrast to oscilloscope measurements in power analysis, the generation of measurement traces is a step-by-step process comparable to a sampling oscilloscope. In every signal loop cycle the detector is switched sensitive only for a very short window in time. Detection events in these detection windows are counted in a corresponding time bin. When enough signal cycles have produced enough count events to overcome residual noise, the detection window is shifted relative to the signal and the process starts again. This process is repeated until the signal has been fully reconstructed. To implement this detection scheme we use an FPGA-based controller phase-locked to the DUT clock. As the DUT executes the target program code, the phase-locked FPGA digitally delays and triggers the APD-detection windows. Detection events are sent back to the FPGA and counted. The measurement time to reconstruct the complete signal can be immense as the number of necessary samples to achieve an adequate signal-to-noise ratio can reach hundreds of thousands. To drastically reduce the measurement times, the FPGA triggers hundreds of gates per signal loop, which results in interleaved measurements.

4.2 Device under Test

For our proof of concept we implemented software AES on a common microcontroller, the ATmega328P. The chip was prepared using a standard automated backside sample preparation machine, commonly used in failure analysis. The substrate was thinned to approximately 50μm, which drastically reduced the exposure time required for emission images. Since silicon is transparent to InGaAs detectors and if the position of the points of emission is otherwise known, this step could even be omitted. In this case, only the IC package needs to be removed, which can be done with standard hand-held rotary tools. The prepared chip was inversely soldered into a cavity on a custom printed circuit board to reduce the working distance to the die surface.

The ATmega328P microcontroller is based on the 8-bit AVR architecture. The AVR architecture is an 8-bit architecture with a 16-bit or 32-bit fetch and 16-bit data memory addresses. In this work we attack the AVR architecture's datapath to recover photonic side channel leakage from the subroutine presented in Figure 3. For this reason it is important to consider several features of the AVR architecture to fully understand the potential attack surface.

The 8-bit registers `r26` and `r27`, `r28` and `r29`, and `r30` and `r31` form the low and high bytes of 16-bit registers `X`, `Y` and `Z`, respectively. On the ATmega328P SRAM is mapped to the data memory and is accessed via load (`ld`)



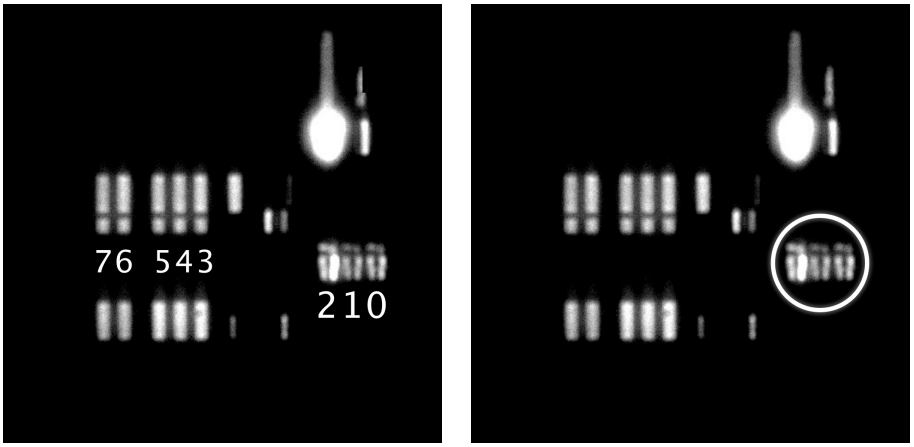
(a) Write access to address 0x300 (b) Read of the value 0xFF

Fig. 1. Emission images of memory accesses on the ATmega328P. The SRAM line at 0x300 is clearly visible in Figure 1(a). Figure 1(b) shows the highlighted area of Figure 1(a) in greater detail. The driving inverters for the first and second SRAM banks are mirrored as is evident in Figure 1(b).

and store (`st`) instructions in conjunction with the registers `X`, `Y` and `Z` for data indirect memory addressing. Load and store operations can also optionally pre- and post- increment or decrement the pointers of the operation. These instructions make it possible to access consecutive bytes of memory without having to reload the pointer. Conditional branches, as well as any load or store operations generally take two clock cycles to execute.

The ATMega328P has four 512-byte memory banks. Each bank is individually connected to the rest of the datapath, see Figures 1(a) and 1(b). This connection consists of very large driving inverters, which are clearly visible in the emission of Figure 1(b). By studying emission images with the techniques introduced in [17], we were able to determine that the emissions are both data and address dependent. The bit order of the emissions could also be determined by analyzing emission images for reads of known values, see Figure 2(a).

Considering the IC's layout, the emissions formed two groups, the five Most Significant Bits (MSB) and the three Least Significant Bits (LSB). Because of the distance between the two groups and the additional enable and clock signals that lie between them, it is impractical to measure the emissions of both groups in a single trace. For this reason we chose to measure the 5 MSB and the 3 LSB separately. Figure 2(a) also clearly shows that the emission of the 3 LSB are dominated by the emissions of b_2. This corresponds to the results of the analysis detailed in Section 5.1. For this reason we chose to use the emissions of just the LSB (b_2, b_1 and b_0) in Section 5.2, see Figure 2(b).



(a) Bit order of the driving inverters (b) Position and aperture for the 3 LSB

Fig. 2. Emission images of the driving inverters for the second SRAM bank on the ATmega328P. Figure 2(a) shows the bit order of the driving inverters. Figure 2(b) shows the three Least Significant Bits and the approximate position and aperture, which was used in subsequent emission traces.

4.3 Software AES

The software AES executed on the microcontroller was identical to the open-source implementation employed in [19] and is freely available at [1]. Figure 3 is the assembly code for the compiled `SubBytes` operation used in the software implementation. As already mentioned in Section 4.2, the conditional branching operations (`brne`) and the load and store operations (`ld` and `st`) each take two clock cycles to execute. The 16 state bytes and the AES S-Box were located in the SRAM of the microcontroller. The AES S-Box was located at the address `0x23F`. In the `SubBytes` function register `X` points to the address of the 16 state bytes. To perform the `SubBytes` operation, a state byte is read (`ld r30,X`). The value of this state byte is the result of the initial `AddRoundKey` operation. Next, this value is used to index the AES S-Box by adding an offset to this value, i.e., the base address of the AES S-Box. The `avr-gcc` compiler uses the subtract operations, `subi` and `sbci`, and the complementary immediate values `0xC1` and `0xFD` because subtract operations are executed in a single clock cycle. The S-Box output is loaded and stored and the `X` pointer is incremented to point to the next state byte. The `cpi` operation ensures that only 16 bytes are actually substituted by the subroutine.

```
1  subBytes:
2    cbi   PORTB, Pin5 ; Set trigger
3    ldi   r24, 0x00   ; i = 0
4  do_subBytes:
5    ld    r30, X      ; Load &state[i]
6    ldi   r31, 0
7    subi  r30, 0xC1   ; Add SBox low address byte (0x3F)
8    sbci  r31, 0xFD   ; Add SBox high address byte (0x02)
9    ld    r25, Z      ; Load &SBox + &state[i]
10   st    X+, r25     ; Store new state[i]
11   subi  r24, 0xFF   ; i++
12   cpi   r24, 16     ; i < 16?
13   brne  do_subBytes
14   sbi   PORTB, Pin5 ; Clear trigger
```

Fig. 3. `SubBytes` Operation

5 Practical Results

In this section, we present the complete DPEA that led to the recovery of the secret key. First, we present an analysis based on the correlation coefficient. Next, we accomplish the DPEA using the Difference of Means method. Both methods show that DPEA also helps to gain knowledge about the attacked device.

As a proof of concept, we attacked AES-128 encryption. Since AES operates on bytes, we attacked and revealed each of the 16 key bytes separately. Therefore, unless otherwise stated, the description in the remainder of this section always refers to a fixed but arbitrary byte. We used each possible value as input data, that is, $\mathcal{D} = \mathcal{K} = \{0, 1, \ldots, 255\}$. The analyzed traces were recorded at the driving inverters for the second SRAM bank. To arrive at an acceptable signal-to-noise ratio, we averaged one million traces for every input value, in the manner described in Section 4.1. Additionally, due to the chip's layout, we got two averaged traces for each $d \in \mathcal{D}$, one for the LSB measurement and one for the MSB measurement. Each of these covers the complete first `SubBytes` operation, which consists of three main instructions, each taking two clock cycles to execute, as described in Section 4. These three main instructions are clearly visible as six dominant peaks in Figure 4. Since a DPEA requires an intermediate result which depends as well on the input data as on the secret key, we chose to analyze the third and fourth of these peaks, i.e., the second instruction. We denote the points in time of the third and fourth peak belonging to the analyzed byte with i_3 and i_4, respectively.

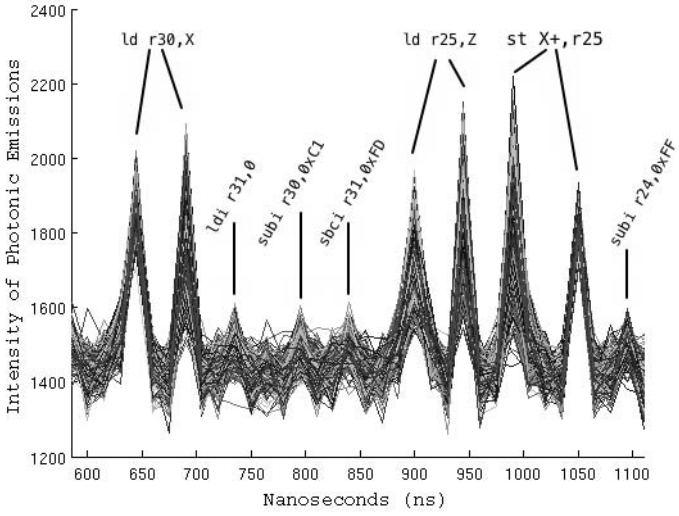


Fig. 4. Emission trace of the memory accesses for a single byte during the `SubBytes` operation. The two cycle memory access instructions, i.e. `ld` and `st`, result in a much higher intensity of photonic emissions as compared to other instructions.

5.1 Correlation Analysis

Our first analysis is strongly related to the DPA using Pearson correlation as means of statistical analysis [14]. This analysis has to be applied to both the MSB and LSB measurements separately. Thus, indeed we have two independent sets of key candidates, $\mathcal{K}_{\mathrm{MSB}} = \{0, \ldots, 31\}$ and $\mathcal{K}_{\mathrm{LSB}} = \{0, \ldots, 7\}$. For simplicity, we will refer to these as just $\mathcal{K}$. For each key hypothesis $k \in \mathcal{K}$, we have two vectors of length D, that we denote $\mathbf{h}_{\mathbf{k},\mathbf{i_3}}$ and $\mathbf{h}_{\mathbf{k},\mathbf{i_4}}$, respectively: The first one's entries are the hypothetical values $h(0, k, i_3)$ to $h(255, k, i_3)$, and the second one's $h(0, k, i_4)$ to $h(255, k, i_4)$. Accordingly, we extracted two 256-entry vectors from the recorded traces: The first one, $\mathbf{t}_{i_3}$, consists of the elements $t_{d,i_3}\ \forall\ d \in \mathcal{D}$ and the second one, $\mathbf{t}_{i_4}$, of all elements $t_{d,i_4}\ \forall\ d \in \mathcal{D}$. Having fixed a certain hypothesis function h and one of the points in time, which we will call i^* from now on, we calculated the correlation coefficient $r \in [-1, 1]$ for the vector extracted from the traces, i.e., $\mathbf{t}_{i^*}$, and each of the corresponding vectors of hypothetical values, i.e., $\mathbf{h}_{\mathbf{k},\mathbf{i}^*} \forall\ \mathbf{k} \in \mathcal{K}$. On condition that the hypothesis is reasonable, wrong key hypotheses will lead to low correlations, whereas the correct key hypothesis leads to the highest correlation and thus, reveals the secret key. Considering the hypothesis function h, we followed several approaches: The HW of the respective values, the HD of the respective values and the values of the preceding suboperation, and the HD of the address of the state byte (which is `0x833` for the first and `0x842` for the 16^{th} byte) and the absolute S-Box address. Also, we considered all possibilities of incorporating only certain bits, e.g., regarding the

$$h(d, k, i_3) = (\texttt{0x23F} + (d \bigoplus k))_2$$
$$h(d, k, i_4) = (\texttt{SubBytes}(d \bigoplus k))_2$$

Fig. 5. The only two hypotheses which distinguish clearly two sets of key candidates. Combining these functions, only one to two candidates for the LSB per key byte are left.

Table 1. Result of the correlation analysis for the LSB of key byte 6. The candidates printed in bold are the only ones which lead to positive correlations for both hypothesis functions. Among these is 111, which is the correct part of the secret key.

absolute S-Box address		S-Box output	
LSB	r	LSB	r
111	0.4624	011	0.0593
001	0.1093	**001**	0.0132
110	0.1024	**111**	0.0117
000	0.0298	101	0.0102
100	-0.0298	000	-0.0046
010	-0.1024	100	-0.0115
101	-0.1093	010	-0.0369
011	-0.4624	110	-0.0414

LSB traces, incorporating only bits b_0 and b_1, only b_1, and so on. We also tried out different weighting factors for different bits, including negative weights, since the measured transistors were inverters.

We tested all these functions for both the MSB and the LSB measurements against two sets of data, recorded with different secret keys. From all these possibilities, only two similar hypothesis functions, applied to the LSB measurements, led to reliable results. These functions are defined as the identity function on bit 2 of the absolute S-Box address and as the identity function on bit 2 of the S-Box output, respectively, see Figure 5.

Both these functions divide sharply between two groups of key candidates: the candidates from the first group lead to positive correlations, whereas the second group consists of those candidates with negative correlations. The secret key always leads to a positive correlation. By taking the intersection of the two groups with positive correlations, only one to two candidates for the LSB of the secret key byte remain, among these the correct LSB, see Table 1, which exemplarily shows the result for the 6^{th} key byte.

Referring to Table 1, neither is it standard to have the highest correlation for the correct candidate, which is 111 in this case, nor to have such a big difference between the two highest correlations (cf. the values belonging to the absolute S-Box address). However, this approach only leads to perfect results for these two hypothesis functions. For the MSB measurements, no clear distinguishing function could be found. The best analysis for the MSB measurements reduced the set of possible key MSB to approximately 4, leaving $4 \cdot 2 = 2^3$ candidates per byte and

thus, approximately $(2^3)^{16} = 2^{48}$ possibilities for the whole 128-bit key. With 2^{48} possibilities, the rest of the AES key could be brute forced. However, the insight about bit 2 allows for another analysis, which perfectly reveals the secret key.

5.2 Difference of Means

The Difference of Means (DoM) method was already used in Kocher's first work on DPA [13] and has since occasionally been used, e.g., [14,16]. In contrast to the correlation analysis, the Difference of Means method belongs to the partition distinguishers [21]. It requires and exploits reliable information of just a single bit to reveal the whole key if a nonlinear function can be attacked. The general approach is to partition for each $k \in \mathcal{K}$ the traces according to the value of a certain bit b_i (i.e., 1-bit partition) after a nonlinear function has been calculated. An attacker partitions for each $k \in \mathcal{K}$ the D traces according to the value of the chosen bit, which is 0 or 1, respectively. For each $k \in \mathcal{K}$, the attacker thus gets two sets of traces, and calculates a mean for both. Afterwards, the attacker computes the difference of these two mean traces - hence the name, Difference of Means. The underlying assumption is that in case a key candidate is wrong, the partition of the two sets is more or less random, so that both mean traces are approximately equal and thus, the difference trace gets drowned out by the noise. However, in case the traces were partitioned according to the correct secret key and the emissions of the weighted bit influence the measurements, there is a significant difference in the two mean traces and thus, their difference trace will exhibit a peak at some point in time. Applying the Difference of Means method to our traces, given that bit 2 is a good discriminator, we get perfect results by analyzing just the LSB measurements.

We applied the method to the S-Box output during the first round. Utilizing the knowledge gained about bit 2 in the preceding correlation analysis, we partitioned the traces according to this very bit. Thus, we partitioned for each $k \in \mathcal{K} = \{0, 1, \ldots, 255\}$ the 256 traces according to the value of bit 2 of the S-Box output after the inital key addition. That is, the traces were sorted depending on the value $(\texttt{SubBytes}(d \bigoplus k))_2$. As can be seen in Figure 6, which shows the difference traces restricted to the `SubBytes` operation of the first three state bytes in the first round, we got perfect results.

Surprisingly, the DoM peaks occur at points in time which we did not foresee: Key byte $n, n \in \{1, \ldots, 15\}$ leads to a distinct peak at the moment of the fifth peak of the three main instructions (cf. Section 5, Figure 4), i.e., the third instruction, belonging to state byte $n + 1$. This shows that DPEA also helps to gain knowledge about the DUT. This knowledge can be used to further improve the attacks and to support reverse engineering. However, that is why, in fact, we only revealed the first 15 key bytes, corresponding to the 15 peaks in Figure 7. Although Figure 7 purports a ghost peak at approximately 3 microseconds (μs), this is not ambiguous: As explained, the 16 subsequent `SubBytes` operations can be identified by a visual inspection of the traces. Each of these operations exhibits six dominant peaks, which can be identified in Figure 4. The huge differences in the DoM traces occur exactly at the times of the respective fifth peak. Thus,

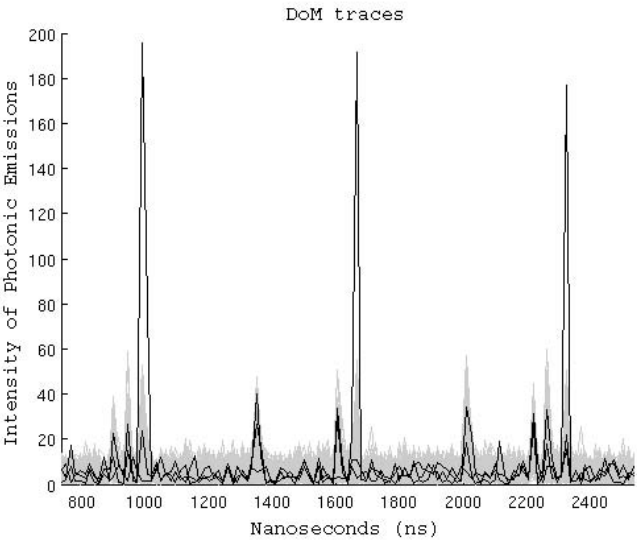


Fig. 6. All 256 DoM traces, between 800 and 2500 nanoseconds. The traces belonging to the first three bytes of an AES-128 key (`0xBD`, `0xDB`, `0xEF`) are plotted in black, whereas the traces belonging to the remaining 253 key candidates are plotted in gray. Each peak corresponds to the relative key byte, and there is no ambiguity, i.e., there are no ghost peaks at relevant points in time.

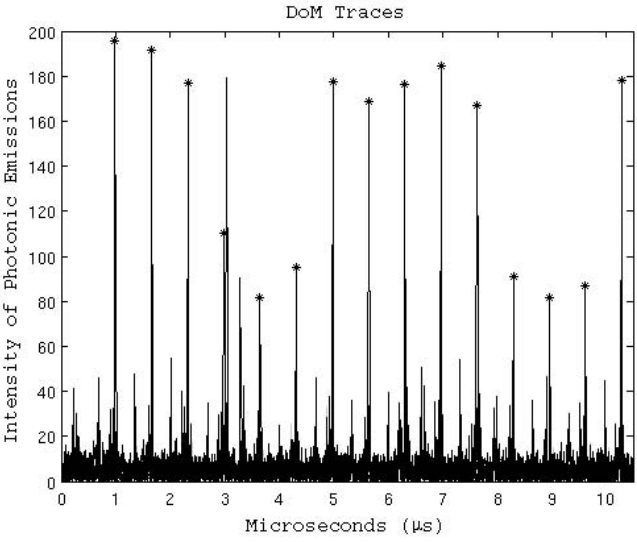


Fig. 7. All 256 DoM traces. The peaks corresponding to the 15 key bytes are clearly marked with a star (*).

it is unambiguous that in Figure 7, regarding the peak corresponding to the fourth key byte, the first peak is the determining peak, although it is lower: The fifth peak of the fourth `SubBytes` operation occurs at 2985 nanoseconds (ns), as does the lower DoM peak, whereas the higher DoM peak, which hence must be wrong, occurs at point 3030 ns. Due to this linkage, the peak at 3285 ns neither raises a disturbance, since it occurs approximately in-between two consecutive fifth peaks.

5.3 Future Work

The suggestions for future work affect the measurement, more sophisticated analysis methods and the development of countermeasures.

As our analysis shows, the emissions of a single transistor are enough to reveal the secret key. Hence, future differential attacks should be based on measurements of a single transistor and thus prove this claim. Referring to our DUT and implementation, this would be the transistor of the driving inverters which corresponds to bit 2. Besides, further analysis methods have to be developed. These can either be based on known methods, e.g., [5,6,21], or directly aim at certain photonic characteristics. Also, higher order attacks have to be developed, cf. [15], suitable to obvious countermeasures like randomization and masking. More importantly, since the photonic side channel poses a serious threat to unprotected implementations, powerful hardware and software countermeasure have to be developed that directly target the leakage from photonic emissions. These can be measures on the technology level, such as absorbing dotant profiles or substrate treatment; the implementation level, such as novel standard cell layouts that reduce data-dependent emission; or photonic side channel specific masking schemes.

6 Conclusion

This work complements the state of the art of Photonic Emission Analysis with the introduction of Differential Photonic Emission Analysis (DPEA). In this work we present the first successful differential analysis of photonic emissions. We were able to recover the full AES-128 secret key by applying differential side channel analysis techniques to the photonic emission measurement. By analyzing emission traces of data-dependent regions of the datapath we were able to recover a single bit of the S-Box output. Subsequently, by applying Difference of Means we were able to recover the full AES secret key. Given its low cost, DPEA proved to be a powerful tool and thus, photonic side channel attacks pose a serious risk to modern security ICs. The extraordinary spatial resolution of this technique and the resulting large number of potentially leaking targets makes successful attacks much more probable. Hardware countermeasures, developed to counter power analysis, can also hinder PEA. However, because emission images allow for a functional understanding of the DUT, most countermeasures can be easily circumvented by selecting a different area. To prevent PEA, countermeasures must be developed to shield photonic emissions from reaching the

observer altogether. Such countermeasures would make the ICs very expensive to produce.

Acknowledgements. The authors acknowledge support by the German Federal Ministry of Education and Research in the project PhotonDA through grant number 01IS10029A and the Helmholtz Research School on Security Technologies. Also, the authors would like to thank our project partners at NXP Semiconductors Germany for their insight and cooperation, the Semiconductor Devices research group at TU Berlin for sample preparation and our colleague Christoph Bayer for helpful discussions and feedback.

References

1. Photon-DA AES Implementation (October 2012), `https://github.com/nedos/pda_aes`
2. Agrawal, D., Archambeault, B., Rao, J.R., Rohatgi, P.: The EM side-channel(s). In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 29–45. Springer, Heidelberg (2003)
3. Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F.X., Veyrat-Charvillon, N.: Mutual information analysis: a comprehensive study. J. Cryptology 24(2), 269–291 (2011)
4. Bernstein, D.: Cache-timing attacks on AES (2004), `http://cr.yp.to/papers.html#cachetiming`
5. Bär, M., Drexler, H., Pulkus, J.: Improved template attacks. In: COSADE (2010)
6. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003)
7. Daemen, J., Rijmen, V.: The design of Rijndael: AES – the Advanced Encryption Standard. Springer, Heidelberg (2002)
8. Di-Battista, J., Courrege, J.-C., Rouzeyre, B., Torres, L., Perdu, P.: When Failure Analysis Meets Side-Channel Attacks. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 188–202. Springer, Heidelberg (2010)
9. Ferrigno, J., Hlaváč, M.: When AES blinks: introducing optical side channel. Information Security, IET 2(3), 94–98 (2008), `http://dx.doi.org/10.1049/iet-ifs:20080038`
10. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, p. 251. Springer, Heidelberg (2001)
11. Heyszl, J., Mangard, S., Heinz, B., Stumpf, F., Sigl, G.: Localized Electromagnetic Analysis of Cryptographic Implementations. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 231–244. Springer, Heidelberg (2012)
12. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
13. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
14. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks – Revealing the Secrets of Smart Cards. Springer (2007)

15. Messerges, T.S.: Using second-order power analysis to attack DPA resistant software. In: Paar, C., Koç, Ç.K. (eds.) CHES 2000. LNCS, vol. 1965, pp. 238–251. Springer, Heidelberg (2000)
16. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining smart-card security under the threat of power analysis attacks. IEEE Trans. Computers 51(5), 541–552 (2002)
17. Nedospasov, D., Schlösser, A., Seifert, J., Orlic, S.: Functional integrated circuit analysis. In: 2012 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST (2012)
18. Quisquater, J.J., Samyde, D.: Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In: E-smart, pp. 200–210 (2001)
19. Schlösser, A., Nedospasov, D., Krämer, J., Orlic, S., Seifert, J.-P.: Simple photonic emission analysis of AES. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 41–57. Springer, Heidelberg (2012)
20. Skorobogatov, S.: Using Optical Emission Analysis for Estimating Contribution to Power Analysis. In: 2009 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pp. 111–119 (2009), http://dx.doi.org/10.1109/FDTC.2009.39
21. Standaert, F.-X., Gierlichs, B., Verbauwhede, I.: Partition *vs.* Comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected CMOS devices. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 253–267. Springer, Heidelberg (2009)
22. Villa, S., Lacaita, A., Pacelli, A.: Photon emission from hot electrons in silicon. Physical Review B 52(15), 10993–10999 (1995), http://www.dx.doi.org/10.1103/PhysRevB.52.10993