Efficient Algorithmic Evaluation of Correlation Power Analysis: Key Distinguisher Based on the Correlation Trace Derivative

Petr Socha, Vojtěch Miškovský, Hana Kubátová, Martin Novotný
Czech Technical University in Prague
Faculty of Information Technology
{sochapet,miskovoj,kubatova,novotnym}@fit.cvut.cz

***Abstract*—Correlation power analysis (CPA) is one of the most common side-channel attacks today, posing a threat to many modern ciphers, including AES. In the final step of this attack, the cipher key is usually extracted by the attacker by visually examining the correlation traces for each key guess. The naïve way to extract the correct key algorithmically is selecting the key guess with the maximum Pearson correlation coefficient.**

We propose another key distinguisher based on a significant change in the correlation trace rather than on the absolute value of the coefficient. Our approach performs better than the standard maximization, especially in the noisy environment, and it allows to significantly reduce the number of acquired power traces necessary to successfully mount an attack in noisy environment, and in some cases make the attack even feasible.

***Index Terms*—Side channel attack, correlation power analysis, Pearson correlation coefficient, key distinguisher, edge detection**

I. Introduction

Side channel attacks (SCAs) pose a serious security threat to many modern cryptographic devices, even those based on ciphers considered mathematically secure, such as AES. One of the most common SCAs today is differential power analysis (DPA) [1] and especially its enhanced, correlation based variant, correlation power analysis (CPA) [2], [3].

The CPA attack is based on measuring the power consumption of a cryptographic device while encrypting random data, and then correlating obtained power traces with the consumption predictions for each key candidate. These predictions are usually based on the knowledge of the cipher implementation and of the random data used. Comparing the Pearson correlation coefficients for different key candidates may give us a correct key candidate. The nature of the CPA attack allows revealing the key in smaller portions, e.g. bytes or nibbles, thus making the whole attack much less computationally demanding than in case of attacking the whole key at once by brute-force.

The number of measured power traces necessary for a successful recovery of the key may be used as a metric for evaluation and comparison of the SCA resistance of various cryptographic systems, alongside other metrics such as success rate, entropy guessing [4] or mutual information analysis [5].

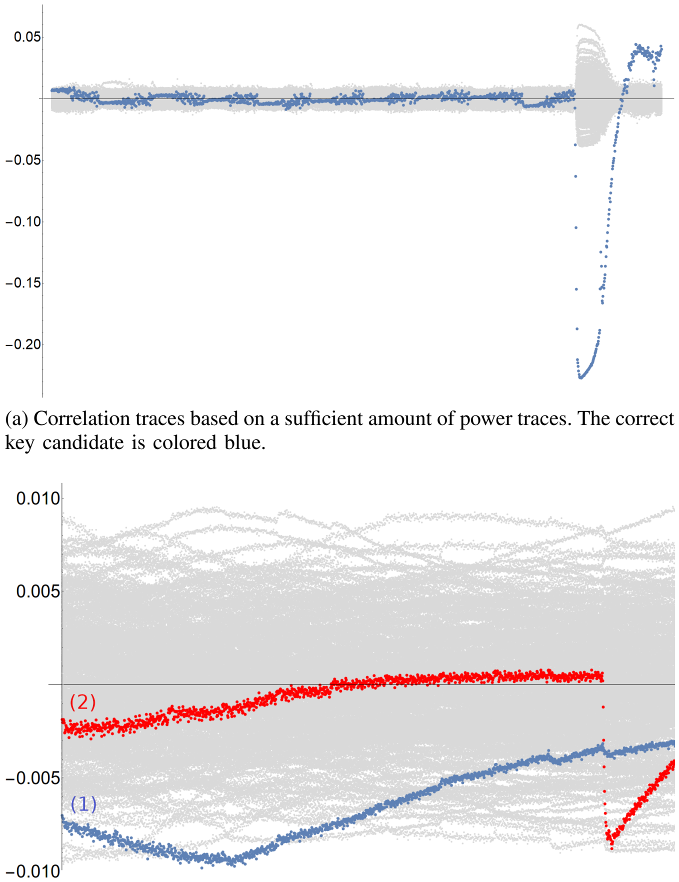
The examination and comparison of correlation traces, in order to obtain a valid key guess, is done visually by the attacker. However, an algorithmization of this final step of the CPA attack is highly relevant for batch attacking and for the automatic evaluation of the side-channel attack. The naïve way to automate this process is simply selecting the key guess which maximizes the Pearson correlation coefficient. In this paper, we propose an algorithmic way of extracting the key guess based on a significant change in the correlation trace rather than on the correlation coefficient magnitude.

II. Related Work

Differential power analysis (DPA), a non-profiled side-channel attack applicable to the implementations of many ciphers such as DES or AES, was introduced in [1], [6]. Different variants of the DPA attack were introduced over the time, such as Correlation power analysis (CPA) [2], [3], which uses Pearson correlation coefficient. Another approach, Mutual information analysis, based on the Shannon's entropy principles, is described in [5]. The Mutual information analysis may be used for the attack itself as well as for leakage assesment.

Different power analysis distinguishers, such as Pearson correlation coefficient or Mutual information, are discussed in [7] regarding their practical usage. The correlation-based methods are recommended in cases, where the attacker is able to derive a good power model predictions. Furthermore, a general statistical model for a side-channel attack analysis, based on the Maximum Likelihood Estimate, is presented in [8]. Many papers, such as [9], discuss and offer solutions for the noise and interference problems when performing the SCAs. Various metrics for the evaluation of the side-channel analysis were published, such as success rate [10] or entropy guessing [4].

Template attacks [11] are example of profiled side-channel analysis, where the assumption is that attacker has the exact same copy of the device under attack, allowing him to create a precise model of the power consumption prior to attacking. The attack itself can be viewed as a classification problem then. In recent years, deep learning techniques are being investigated in context of profiled side-channel analysis [12], including use of multi-layer perceptron and convolutional neural networks [13]. Even though that deep learning approach still suffers from many problems [14], [15], it seems to be



(a) Correlation traces based on a sufficient amount of power traces. The correct key candidate is colored blue.

(b) Correlation traces based on an insufficient amount of power traces. Searching for a (negative) maximum correlation coefficient leads us to the wrong key candidate, which is colored blue (1). The correct key candidate is colored red (2).

Figure 1. Correlation traces (a time series of a Pearson correlation coefficient during the encryption), for all 256 key candidates.

a promising direction. Use of deep learning in non-profiled attack scenarios is discussed in [16].

III. Our Contribution

Our approach extends Correlation power analysis (CPA), i.e. non-profiled side-channel attack, and it is based on detecting a sudden change (edge) in a correlation trace (a time series of a correlation coefficient). With this approach we are able to

- significantly *reduce* the number of acquired power traces necessary to successfully mount an attack in noisy environment, and
- in some cases make the attack even *feasible*.

Reduction of the number of power traces reduces both the acquisition (measurement) time and the time of analysis.

In [17] and [18], both the theory and computational approach to the edge detection and necessary (pre-)processing steps are presented, making use of convolution operation.

IV. CPA Attack Evaluation

Our primary research focus in this paper is AES-128, a block cipher commonly used in many hardware cryptosystems. Since AES implements 8-bit S-Boxes, attacking a byte of the key at a time is possible [3]. We are able to predict the power consumption of the device when encrypting/decrypting a certain plain/cipher text, and since there are only $2^8 = 256$ possibilities for a byte of the key, comparing a real power consumption with our power predictions is computationally acceptable. Since we do not know the exact time when the predicted values correlate, we measure the consumption during the whole encryption (or a specific part of it), giving us a finite number of samples.

We call this collection of samples, obtained during a single encryption, a power trace. In the first phase of the attack, a set of power traces is measured. Correlating our 256 predictions (for every power trace measured) with real power consumption at each sample point gives us 256 different time series of a Pearson correlation coefficient, which we call correlation traces. These can be seen in Figure 1.

A. Motivation

To evaluate the correlation analysis results, the attacker would visually examine the correlation traces, looking for specific anomalies that might give him a hint about the right cipher key. The naïve way to automate this process might be searching for the correlation curve with the strongest (positive or negative) correlation and thus relevant key candidate.

Correlating our predictions with each sample point in the power trace gives us a correlation matrix with dimensions $m \times 256$, with m being the number of samples per trace. Looking for the maximum Pearson correlation coefficient in this matrix gives us a hint for selecting the correct key candidate. In situation depicted in Figure 1a, this approach works just fine.

However, the shape of the correlation curve in time is still more informative, than the magnitude of the correlation coefficient itself. In Figure 1b, one can easily identify the correct key candidate by the naked eye (red curve (2)), while looking for the Pearson correlation coefficient with the highest absolute value fails, as in such a case the blue curve (1) would be selected. Note that with more measurements and power traces available, the spike on the red curve (2) would grow bigger, while correlation at other samples would converge to zero.

The algorithmic evaluation of the correlation coefficients using the naïve maximum likelihood-based method may become even more problematic when there is a significant noise present. Correlation trace for the correct key candidate can be seen in Figure 2. This trace was obtained from a board featuring a switching power supply, which represents a significant source of a background noise. Identifying the key candidate by the naked eye is possible, but time demanding, and searching for a maximum/minimum correlation coefficient does not work well. Our observations led us to the idea of examining the *progression* of the correlation coefficient in time to algorithmically give the attacker a hint about the correct key guess, rather than examining its *instantaneous magnitude*. According to our research, when correlated working variable causes a change in the power consumption of the device, an edge typically appears in the correlation trace. This problem

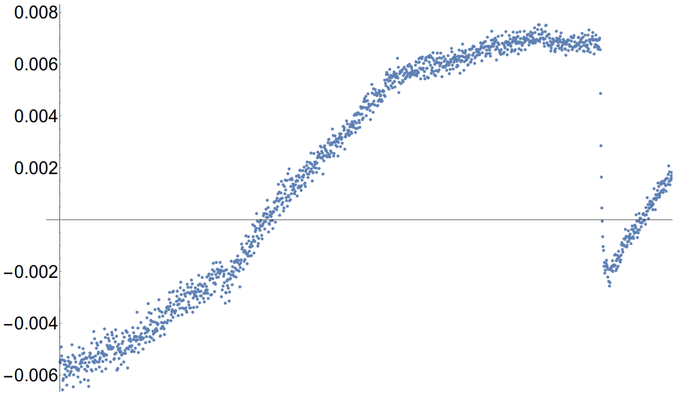


Figure 2. Correlation trace obtained while attacking AES on DPABoard [19] (Artix 7 FPGA Board with a switching power supply).

is very similar to image edge detection problem as described in [17], [18].

Since edge detecting operators are very sensitive to noise, appropriate filtering/smoothing of the correlation traces must be done first. In Subsection IV-B we discuss filters that can be used for smoothing the correlation traces. Subsection IV-C discusses edge detectors explored in this paper. In Subsection IV-D we describe how to combine filters and edge detecting operators into one operation in order to reduce the computational complexity.

B. Noise Filtering/Smoothing

In image processing, typical choice is a Gaussian filtering [18], [17]. For our further experimental purposes, we have chosen two filters: the Moving average filter and the Gaussian filter.

Moving average filter is defined as follows: Assume that $f(t)$ is a discrete variable, then convolution

$$(f * ma(d))(t) = \frac{1}{d} \sum_{i=t-\lfloor \frac{d}{2} \rfloor}^{t+\lceil \frac{d}{2} \rceil -1} f(i) \tag{1}$$

is the result of filtering the variable $f(t)$ using Moving average filter with diameter d.

Gaussian filter is defined as follows: Assume that $f(t)$ is a discrete variable, then convolution

$$(f * g(d,\sigma))(t) = \sum_{i=t-\lfloor \frac{d}{2} \rfloor}^{t+\lceil \frac{d}{2} \rceil -1} f(i) \cdot \frac{\exp(-\frac{(i-t)^2}{\sigma^2})}{\mathrm{norm}(d,\sigma)} \tag{2}$$

is the result of filtering the variable $f(t)$ using Gaussian filter with diameter d and deviation σ, where

$$\mathrm{norm}(d,\sigma) = \sum_{j=-\lfloor \frac{d}{2} \rfloor}^{\lceil \frac{d}{2} \rceil -1} \exp(-\frac{j^2}{\sigma^2}) \tag{3}$$

is the normalization, making sure that the sum of used Gaussian filter equals to 1.

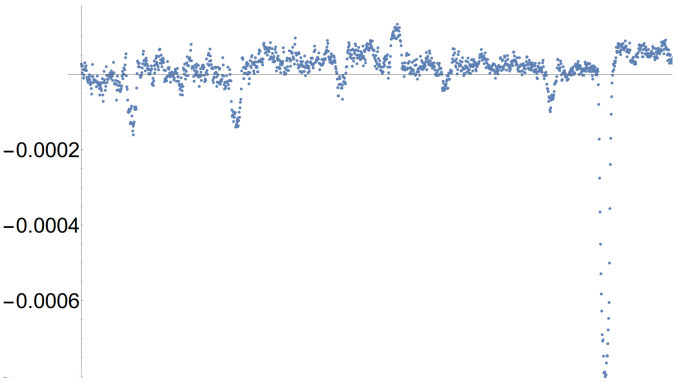


Figure 3. Correlation trace from Figure 2 processed by First derivative operator with Gaussian filtering.

C. Edge Detection

After the noise filtering, the edge detection takes place. There are two approaches to this: a first-derivative based operator, and a second-derivative based/Laplace operator [17].

When the first derivative approach is used, the filtered correlation traces are processed with the first derivative operator, and then the search for the largest absolute value of the derivative is done. When using the second derivative approach, the algorithm searches for significant zero-crossings of the Laplacian of the correlation trace. Both approaches are compared in Section V.

D. Computational Approach

As suggested in [18], both derivative operators and filtering are performed using a discrete convolution. The Moving average filter with diameter d can be easily implemented as a convolutional kernel:

$$ma(d) = \frac{1}{d}[\underbrace{1, 1, \dots, 1}_{d\times}]. \tag{4}$$

In a case of the Gaussian filter with deviation σ, appropriate convolutional kernel of width d can be obtained using a formula:

$$G(x,\sigma) \propto \exp(-\frac{x^2}{\sigma^2}), \tag{5}$$

and making sure, that the sum of all the terms in the kernel is equal to 1. This can be done by dividing every term of the kernel by the sum of all the kernel terms. For example, Gaussian kernel $g(d=5, \sigma=1)$ looks like

$$g(5,1) = [0.06135, 0.2448, 0.3877, 0.2448, 0.06135]. \tag{6}$$

For the approximation of the first derivative, the following convolutional kernel is used:

$$d1 = [-1, 0, 1], \tag{7}$$

while for the approximation of the second derivative, the discrete Laplace kernel is used:

$$d2 = [1, -2, 1]. \tag{8}$$

Thanks to the associativity of convolution, the smoothing and derivative operator can be precomputed beforehand, resulting in one kernel performing both operations at once. First derivative Gaussian convolution kernel can be obtained using formula

$$G(x,\sigma)' \propto \frac{x}{\sigma^2} \cdot \exp(-\frac{x^2}{\sigma^2}), \quad (9)$$

and Laplacian of Gaussian convolution kernel can be obtained using formula

$$\Delta\, G(x,\sigma) \propto \frac{x^2-\sigma^2}{\sigma^4} \cdot \exp(-\frac{x^2}{\sigma^2}). \quad (10)$$

The edge detection on a correlation trace can now be done as a convolution, with time complexity $\mathcal{O}(m \times d)$, where m is the number of samples in the correlation trace, and d is the diameter of the filter.

Searching for the key guess in the correlation matrix consists of applying this convolution on each row of the matrix and looking for the largest value (in case of first derivative) or zero-crossings (in case of Laplacian) in the resulting matrix.

V. Experimental results

We have executed two classes of experiments:

- First, we have evaluated all proposed distinguishers regarding the amount of correctly revealed bytes of the AES-128 cipher key, for a various fixed numbers of power traces available. Results of this class of experiments are presented in Subsection V-A.
- Second, we have evaluated the First derivative + Gaussian distinguisher regarding the filter parameters (filter diameter and deviation). Results of this class of experiments are presented in Subsection V-B.

The platforms we used to evaluate presented methods were following:

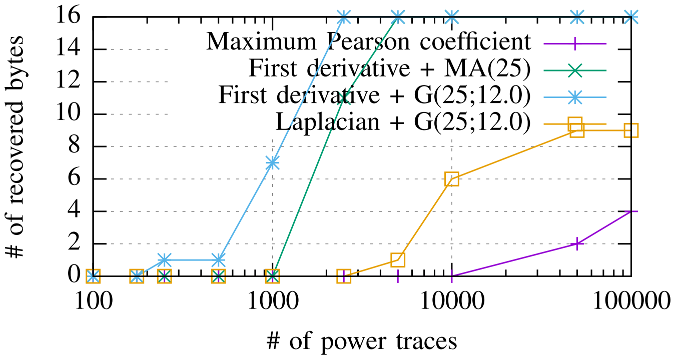
- **DPABoard** [19] (open experimental board) with Xilinx Artix 7 FPGA in two revisions: with a switching power supply, and with a low-noise power supply,
- **Sakura-G** board [20] with Xilinx Spartan 6 FPGA,
- **Evariste III** system [21] with development board containing Altera Cyclone III FPGA, customized by removing the decoupling capacitors.

While working with the DPABoard [19] with a switching power supply, we have experienced a lot of unwanted noise in the measured power traces. A correlation trace based on these power traces is shown in Figure 2. The correlation trace proccesed with First derivative operator is shown in Figure 3.

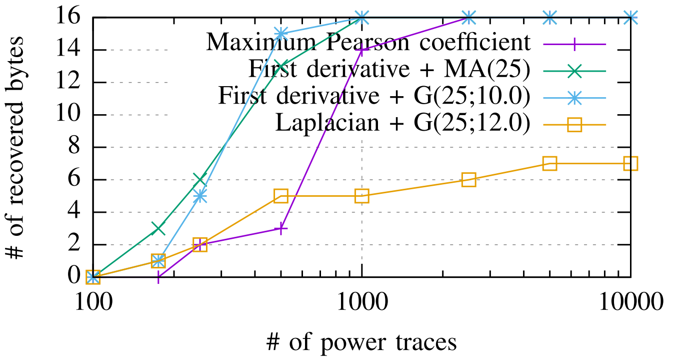
A. Evaluation of Proposed Distinguishers

We evaluated following four distinguishers:

1) standard CPA, maximizing the Pearson correlation coefficient,
2) maximizing the first derivative of correlation traces, smoothed using Moving average,
3) maximizing the first derivative of correlation traces, smoothed using Gaussian filter,
4) searching for zero-crossings of the Laplacian of correlation traces, smoothed using Gaussian filter.



(a) DPABoard (Xilinx Artix 7 FPGA), powered by a switching power supply.



(b) DPABoard (Xilinx Artix 7 FPGA), powered by a low-noise power supply.

Figure 4. Number of succesfully recovered bytes of AES-128 cipher key using different distinguishers, for various number of power traces.

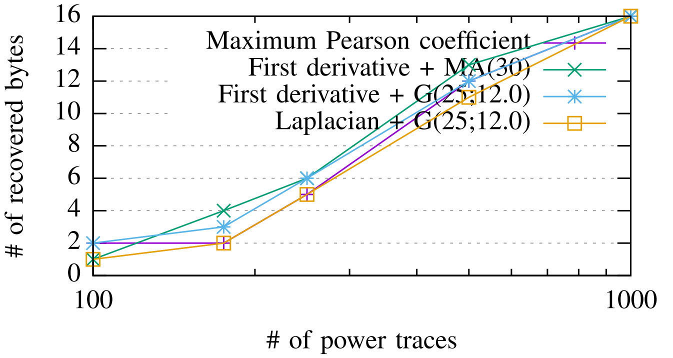


Figure 5. Number of succesfully recovered bytes of AES-128 cipher key using different distinguishers, for various number of power traces, using Sakura-G.

The distinguishers were tested on an AES cipher with 128-bit key, run on three platforms mentioned above. Results are summarized in Tables I-IV. Each table contains number of succesfully recovered key bytes for numbers of power traces varying between 100 and 100,000.

Tables I and II contain the results based on the correlation traces obtained from an open DPA evaluation board DPABoard [19] with Xilinx Artix 7. We have evaluated these distinguishers using two different revisions of the board: Table I and Figure 4a present the results when using the DPABoard with a switching power supply. Table II and Figure 4b present the results when using the DPABoard with

Table I
Number of Correctly Guessed Bytes of the Key, Xilinx Artix 7 With a Switching Power Supply.

# of power traces available **Evaluation method**	100	175	250	500	1k	2.5k	5k	10k	50k	100k
Maximum Pearson correlation coefficient	0	0	0	0	0	0	0	0	2	4
First derivative + Moving Average (d=25)	0	0	0	0	0	11	16	16	16	16
First derivative + Gaussian (d=25, σ=12.0)	0	0	1	1	7	16	16	16	16	16
Laplacian of Gaussian (d=25, σ=12.0)	0	0	0	0	0	0	1	6	9	9

Table II
Number of Correctly Guessed Bytes of the Key, Xilinx Artix 7 With a Low-Noise Power Supply.

# of power traces available **Evaluation method**	100	175	250	500	1k	2.5k	5k	10k	50k	100k
Maximum Pearson correlation coefficient	0	0	2	3	14	16	16	16	16	16
First derivative + Moving Average (d=25)	0	3	6	13	16	16	16	16	16	16
First derivative + Gaussian (d=25, σ=10.0)	0	1	5	15	16	16	16	16	16	16
Laplacian of Gaussian (d=25, σ=12.0)	0	1	2	5	5	6	7	7	7	7

Table III
Number of Correctly Guessed Bytes of the Key, Sakura-G (Xilinx Spartan 6 With a Low-Noise Power Supply).

# of power traces available **Evaluation method**	100	175	250	500	1k	2.5k	5k	10k	50k	100k
Maximum Pearson correlation coefficient	2	2	5	12	16	16	16	16	16	16
First derivative + Moving Average (d=30)	1	4	6	13	16	16	16	16	16	16
First derivative + Gaussian (d=25, σ=12.0)	2	3	6	12	16	16	16	16	16	16
Laplacian of Gaussian (d=25, σ=12.0)	1	2	5	11	16	16	16	16	16	16

Table IV
Number of Correctly Guessed Bytes of the Key, Evariste III + Altera Cyclone III With a Low-Noise Power Supply.

# of power traces available **Evaluation method**	100	175	250	500	1k	2.5k	5k	10k	50k	100k
Maximum Pearson correlation coefficient	0	2	5	12	16	16	16	16	16	16
First derivative + Moving Average (d=25)	2	4	6	10	16	16	16	16	16	16
First derivative + Gaussian (d=25, σ=10.0)	2	3	4	11	16	16	16	16	16	16
Laplacian of Gaussian (d=25, σ=10.0)	0	1	1	2	2	2	6	8	11	11

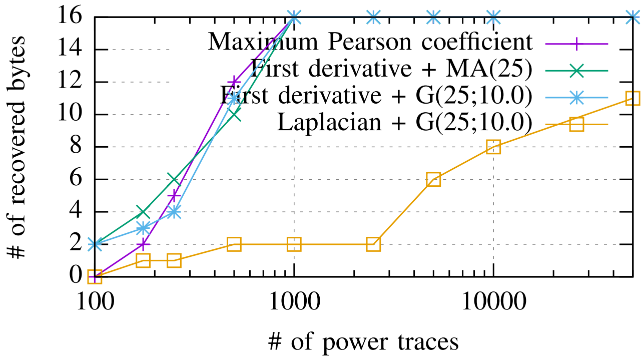


Figure 6. Number of succesfully recovered bytes of AES-128 cipher key using different distinguishers, for various number of power traces, using Evariste III + Altera Cyclone III.

a low-noise power supply.

As can be seen in Figure 4a, in case of noisy power traces obtained from the board with a switching power supply, the performance of both First derivative based distinguishers is much better than the performance of Maximum Pearson correlation coefficient method, which actually fails. While in case of First derivative approach we needed just 2,500 power traces to succesfully reveal all 16 bytes of the key, Maximum Pearson correlation coefficient method did not reveal any byte of the key with the same amount of power traces, and only 4 bytes with 100,000 power traces available.

Figure 4b presents the number of successfully recovered bytes of the key at Xilinx Artix 7 platform with a low-noise power supply. As can be seen, even in noiseless environment, our method provides better results. While in case of First derivative approach we needed just 1,000 power traces to successfully reveal all 16 bytes of the key, in case of Maximum Pearson correlation coefficient method we needed 2,500 traces

Table V
NUMBER OF POWER TRACES NECESSARY TO OBTAIN A FULL AES ENCRYPTION KEY (16 BYTES), RUNNING ON DPABOARD (XILINX ARTIX 7 WITH A SWITCHING POWER SUPPLY).

Maximum Pearson corr. coef.	>100,000

First Derivative + Gaussian (various parameter settings)									
diameter (d) → ↓ **deviation** (σ)	5	11	17	23	29	35	41	47	53
1.0	13,400	13,400	13,400	13,400	13,400	13,400	13,400	13,400	13,400
2.0	9,400	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500
4.0	9,400	4,600	4,500	4,400	4,400	4,400	4,400	4,400	4,400
8.0	9,400	4,000	3,700	2,400	2,400	2,400	2,400	2,400	2,400
12.0	9,500	3,900	3,100	2,400	2,400	2,400	2,400	2,400	2,400
16.0	9,500	3,900	3,100	2,400	2,400	2,400	2,400	2,400	2,400
20.0	9,500	3,900	3,100	2,400	2,400	2,400	2,400	2,400	2,800
24.0	9,500	3,900	3,100	2,400	2,400	2,500	2,400	2,400	2,800
30.0	9,500	3,900	3,100	2,400	2,400	2,500	2,400	2,800	2,800

to fully recover the whole key. The Laplacian of Gaussian distinguisher did not prove to be any more effective than the standard CPA. This may be due to the higher noise sensitivity of the second derivative approach.

Table III and Figure 5 present the results obtained while using Sakura-G [20] board, equipped with Xilinx Spartan 6 chip and a low-noise (linear) power supply. In this case, all methods perform similar, although the first derivative based distinguishers provide a slightly better results when there is insufficient amount of power traces available.

Table IV presents the results for the Evariste III [21] with Altera Cyclone III FPGA and a low-noise (linear) power supply. In this case, first derivative distinguishers and standard CPA are comparable again. First derivative approach may perform a little better for a low amount of power traces, nevertheless, at least 1,000 power traces were necessary for a recovery of the full key. The Laplacian of Gaussian operator did not prove to be any useful in this case either, as can be seen in Figure 6.

B. Gaussian Parameters Evaluation

Previous results, summarized in Tables I-IV, indicate the First derivative distinguisher with Gaussian filtering to be the most promising method. It is particularly successful in noisy environment, as demonstrated in Table I and Figure 4a.

The performance of the edge detecting operator depends on the selection of its smoothing filter parameters:

- diameter of the filter (d), and
- deviation of the Gaussian (σ).

In this section, we evaluate the First derivative distinguisher with Gaussian filtering, using all the evaluation platforms listed in the previous subsection, and compare the results with the Maximum Pearson correlation coefficient method.

Table V presents a number of power traces necessary for obtaining the whole 16-byte long AES key, using the open FPGA evaluation platform DPABoard [19] (Xilinx Artix 7) with a switching power supply. While for the Maximum Pearson distinguisher, we could not retrieve the whole key even with 100,000 power traces available, with optimal First derivative distinguisher only 2,400 power traces are necessary to obtain the whole key.

Table VI presents results obtained using the same evaluation platform, but equipped with the low-noise power supply. In this case, less than a half of power traces is necessary for a successful attack when using First derivative distinguisher compared to the standard CPA Maximum Pearson correlation coefficient (500 vs 1,200).

Table VII presents results obtained when using the Sakura-G [20] evaluation platform (Xilinx Spartan 6), where the performance of both distinguishers is similar.

Table VIII presents the results obtained using the Evariste III [21] + Altera Cyclone III platform. In this case, the optimal First Derivative distinguisher performs better than the Maximum Pearson method.

As can be seen from results summarized in Tables V-VIII, the selection of the filter parameters (d, σ) is crucial for a satisfactory performance of a distinguisher with Gaussian filtering. In our case, the deviation σ should be $\sigma \geq 8.0$ to minimize the number of power traces necessary for succesful identification of a correct key. The diameter d also influences the success of the method, although its impact is not that strong as in the case of deviation. Nevertheless, the diameter d should be large enough to fit the Gaussian with selected deviation, in our case $d \geq 23$.

VI. CONCLUSION

We have presented a new algorithmic approach to the final step of the CPA attack, which is a selection (distinguishment) of the correct key guess from the correlation traces.

Selecting the key candidate which maximizes the correlation coefficient, according to the maximum likelihood principle, is quite sufficient if the cryptographic device runs in an environment well suitable for power trace measurements. However, this method may fail with presence of noise or interference present in the production environment, caused e.g. by a switching power supply.

We show that our distinguisher based on first derivative edge detection is more successful when evaluating the correlation

Table VI
NUMBER OF POWER TRACES NECESSARY TO OBTAIN A FULL AES ENCRYPTION KEY (16 BYTES), RUNNING ON DPABOARD (XILINX ARTIX 7 WITH A LOW-NOISE POWER SUPPLY).

Maximum Pearson corr. coef.	1,200								
First Derivative + Gaussian (various parameter settings)									
diameter (d) → ↓ **deviation** (σ)	5	11	17	23	29	35	41	47	53
1.0	>20,000	>20,000	>20,000	>20,000	>20,000	>20,000	>20,000	>20,000	>20,000
2.0	11,300	3,800	3,800	3,800	3,800	3,800	3,800	3,800	3,800
4.0	10,500	600	600	600	600	600	600	600	600
8.0	9,900	700	600	500	500	500	500	500	500
12.0	9,900	900	600	500	500	500	500	500	500
16.0	9,900	900	600	500	500	600	600	600	600
20.0	9,900	900	600	500	500	500	600	600	600
24.0	9,900	900	600	500	500	500	600	600	600
30.0	9,900	900	600	500	500	500	600	600	600

Table VII
NUMBER OF POWER TRACES NECESSARY TO OBTAIN A FULL AES ENCRYPTION KEY (16 BYTES), RUNNING ON SAKURA-G (XILINX SPARTAN 6 WITH A LOW-NOISE POWER SUPPLY).

Maximum Pearson corr. coef.	800								
First Derivative + Gaussian (various parameter settings)									
diameter (d) → ↓ **deviation** (σ)	5	11	17	23	29	35	41	47	53
1.0	>20,000	>20,000	>20,000	>20,000	>20,000	>20,000	>20,000	>20,000	>20,000
2.0	6,200	2,200	2,200	2,200	2,200	2,200	2,200	2,200	2,200
4.0	5,400	1,200	1,200	1,200	1,200	1,200	1,200	1,200	1,200
8.0	5,400	1,200	1,000	900	900	900	900	900	900
12.0	5,400	1,200	900	900	900	800	800	800	800
16.0	5,400	1,200	900	900	900	800	800	700	700
20.0	5,400	1,200	900	900	800	800	700	700	700
24.0	5,400	1,200	900	900	800	800	700	700	700
30.0	5,400	1,200	900	900	800	800	700	700	700

Table VIII
NUMBER OF POWER TRACES NECESSARY TO OBTAIN A FULL AES ENCRYPTION KEY (16 BYTES), RUNNING ON EVARISTE III + ALTERA CYCLONE III WITH A LOW-NOISE POWER SUPPLY.

Maximum Pearson corr. coef.	700								
First Derivative + Gaussian (various parameter settings)									
diameter (d) → ↓ **deviation** (σ)	5	11	17	23	29	35	41	47	53
1.0	6,800	6,800	6,800	6,800	6,800	6,800	6,800	6,800	6,800
2.0	2,100	1,400	1,400	1,400	1,400	1,400	1,400	1,400	1,400
4.0	1,900	900	800	800	800	800	800	800	800
8.0	9,900	900	900	700	700	700	700	700	700
12.0	9,900	900	900	900	700	700	700	700	700
16.0	9,900	900	900	900	700	700	700	600	600
20.0	9,900	900	900	900	800	700	700	700	700
24.0	9,900	900	900	900	800	800	900	700	700
30.0	9,900	900	900	900	800	900	900	700	700

traces obtained in noisy environment, such as that made by the switching power supplies. Using our method, approximately 2,400 power traces were necessary for a recovery of the whole key, while maximization of Pearson correlation coefficient failed to do so even with 100,000 power traces.

While working with low-noise/linear power supplies and having a sufficient amount of power traces available, both approaches work equally good. When the amount of power traces is insufficient, our first derivative method may provide slightly better results as well. The Laplacian of Gaussian based distinguisher did not prove to be much useful.

The extra time complexity of proposed methods is insignificant compared to the rest of the CPA attack. The resulting reduction of the power traces necessary to reveal the cipher key is even more beneficial considering that the measuring of the power traces is by far the most time consuming part of the attack. Although the time complexity of distinguishers with Gaussian filtering increases with incresing diameter d, the (very slight) increase of time is more than compensated by reducing the time necessary for both acquisition of power traces (i.e. measurement by an oscilloscope) and for calculation of correlation traces.

Acknowledgment

This research has been partially supported by the CELSA project "DRASTIC: Dynamically Reconfigurable Architectures for Side-channel analysis protection of Cryptographic implementations" (CELSA/17/033); the grant GA16-05179S of the Czech Grant Agency "Fault-Tolerant and Attack-Resistant Architectures Based on Programmable Devices: Research of Interplay and Common Features" (2016-2018); and CTU project SGS17/213/OHK3/3T/18.

References

[1] P. Kocher, J. Jaffe, and B. Jun, *Differential Power Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[2] B. den Boer, K. Lemke, and G. Wicke, "A dpa attack against the modular reduction within a crt implementation of rsa," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2002, pp. 228–243.

[3] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2004, pp. 16–29.

[4] F.-X. Standaert, T. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks." in *Eurocrypt*, vol. 5479. Springer, 2009, pp. 443–461.

[5] N. Veyrat-Charvillon and F.-X. Standaert, "Mutual information analysis: How, when and why?." in *CHES*, vol. 5747. Springer, 2009, pp. 429–443.

[6] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Investigations of power analysis attacks on smartcards." *Smartcard*, vol. 99, pp. 151–161, 1999.

[7] E. Oswald, L. Mather, and C. Whitnall, "Choosing distinguishers for differential power analysis attacks," in *Non-Invasive Attack Testing Workshop*, 2011, pp. 1–14.

[8] Y. Fei, A. A. Ding, J. Lao, and L. Zhang, "A statistics-based fundamental model for side-channel attack analysis." *IACR Cryptology ePrint Archive*, vol. 2014, p. 152, 2014.

[9] W. Liu, L. Wu, X. Zhang, and A. Wang, "Wavelet-based noise reduction in power analysis attack," in *Computational Intelligence and Security (CIS), 2014 Tenth International Conference on*. IEEE, 2014, pp. 405–409.

[10] F.-X. Standaert, P. Bulens, G. de Meulenaer, and N. Veyrat-Charvillon, "Improving the rules of the dpa contest." *IACR Cryptology ePrint Archive*, vol. 2008, p. 517, 2008.

[11] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2002, pp. 13–28.

[12] L. Lerman, R. Poussier, O. Markowitch, and F.-X. Standaert, "Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version," *Journal of Cryptographic Engineering*, vol. 8, no. 4, pp. 301–313, 2018.

[13] E. Prouff, R. Strullu, R. Benadjila, E. Cagli, and C. Dumas, "Study of deep learning techniques for side-channel analysis and introduction to ascad database." *IACR Cryptology ePrint Archive*, vol. 2018, p. 53, 2018.

[14] Y. Zhou and F.-X. Standaert, "Deep learning mitigates but does not annihilate the need of aligned traces and a generalized resnet model for side-channel attacks," *Journal of Cryptographic Engineering*, pp. 1–11, 2019.

[15] S. Picek, A. Heuser, A. Jovic, S. Bhasin, and F. Regazzoni, "The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018.

[16] B. Timon, "Non-profiled deep learning-based side-channel attacks." *IACR Cryptology ePrint Archive*, vol. 2018, p. 196, 2018.

[17] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[18] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[19] M. Bartík and J. Buček, "A low-cost multi-purpose experimental fpga board for cryptography applications," in *Advances in Information, Electronic and Electrical Engineering (AIEEE), 2016 IEEE 4th Workshop on*. IEEE, 2016, pp. 1–4.

[20] H. Guntur, J. Ishii, and A. Satoh, "Side-channel attack user reference architecture board sakura-g," in *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on*. IEEE, 2014, pp. 271–274.

[21] N. Bochard, C. Marchand, O. Pet'ura, L. Bossuet, and V. Fischer, "Evariste iii: A new multi-fpga system for fair benchmarking of hardware dependent cryptographic primitives," in *Workshop on Cryptographic Hardware and Embedded Systems, CHES 2015*, 2015.