Correlation Power Analysis 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. The simplest method to extract the correct key guess is selecting the guess with the maximum Pearson correlation coefficient. We propose another 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 CPA, especially in the noisy environment.**

***Index Terms*—Side channel attack, AES, 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 correlation coefficients for different key candidates and selecting the one with the strongest correlation gives us a 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.

In this paper, we propose a different 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 side channel attack applicable to the implementations of many ciphers such as DES or AES, was introduced in [1], [4]. Different variants of the DPA attack were introduced over the time, one of them being the Correlation power analysis (CPA) [2], [3], using Pearson correlation coefficient.

Differential power analysis distinguishers are discussed e.g. in [5]. Various metrics for the evaluation of the attack were published, such as success rate [6], entropy guessing [7] or mutual information analysis [8]. A statistical model for a side channel attack analysis is presented in [9]. Many papers, such as [10], deal with the noise and interference problems when performing the SCAs.

Our approach is based on detecting a sudden change (edge) in a correlation trace (a time series of a correlation coefficient). In [11] and [12], both the theory and computational approach to the edge detection are presented.

III. CPA Attack Evaluation

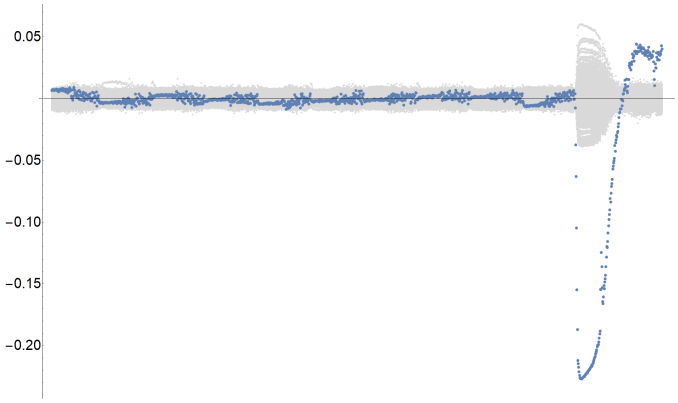
Our primary research focus in this paper is AES-128, a block cipher commonly used in many hardware cryptosystems. Since AES implements an 8 bit S-Boxes, attacking a byte of the key at a time is possible [3]. We are able to predict the power consumption of the device when encrypting/decrypting a certain 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 predictions is computationally acceptable. Since we do not know the exact time when the predicted values correlate, we need to measure the consumption during the whole encryption, giving us a finite number of samples.

We call this collection of samples, obtained during a single encryption, a power trace. Correlating our 256 predictions 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.

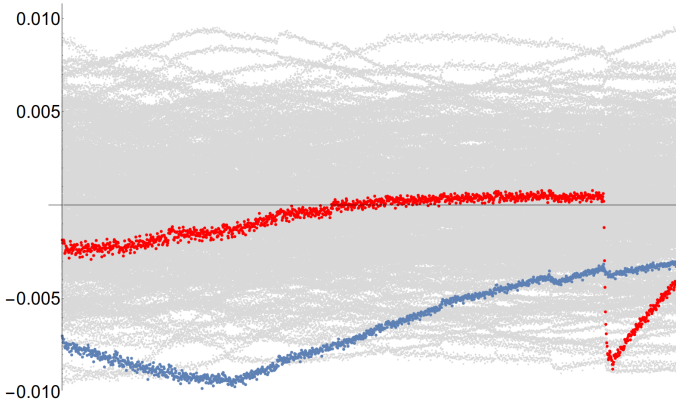
A. Motivation

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 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, while looking for the Pearson correlation coefficient with the highest absolute value fails. With more measurements and power traces available, the spike on the red curve would grow bigger, while other samples would converge to zero.



(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. The correct key candidate is colored red.

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

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 is very similar to image edge detection problem as described in [11], [12].

Since these edge detecting operators are very sensitive to noise, appropriate filtering/smoothing of the correlation traces must be done first.

B. Noise Filtering/Smoothing

For our futher 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.

C. Edge Detection

After the noise filtering, the edge detection takes place. There are two approaches to this: a first-derivative based and a second-derivative based [11].

When the first derivative approach is used, the filtered correlation traces are processed with the first derivative operator. Searching for the maximum/minimum value in these processed correlation traces works well as a CPA key distinguisher, as presented in Section IV. 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 IV.

The discrete derivative operators, as well as the implementations of the filters, are described in the following Subsection.

D. Computational Approach

As suggested in [12], 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:

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

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:

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

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	1	1	7	16	16	16	16	16
Laplacian of Gaussian (d=25, σ=12)	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 LINEAR 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	1	5	15	16	16	16	16	16	16
Laplacian of Gaussian (d=25, σ=12)	0	1	2	5	5	6	7	7	7	7

Thanks to the associativity of convolution, the smoothing and derivative operator can be precomputed beforehand, resulting in one kernel performing both operations at once. When filtering using Gaussian, these kernels (first derivative, Laplacian) can be obtained using following formulas:

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

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

$\Delta\, G(x,\sigma)$ is also known as Laplacian of Gaussian.

The edge detection on a correlation trace can now simply be done as a convolution, with time complexity $\mathcal{O}(m \times d)$, where m is 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.

IV. EXPERIMENTAL RESULTS

We have evaluated proposed distinguishers regarding the amount of correctly revealed bytes of the AES-128 cipher key. The platforms we used to evaluate presented methods were following:

- **DPABoard** [13] (open experimental board) with Xilinx Artix 7 FPGA in two revisions: with a switching power supply, and with a linear power supply,
- **Sakura-G** board [14] with Xilinx Spartan 6 FPGA,
- **Evariste III** system [15] with development board containing Altera Cyclone III FPGA, customized by removing the decoupling capacitors.

A. DPABoard (Xilinx Artix 7)

Tables I and II contain the number of succesfully recovered bytes of the cipher key. The processed correlation traces were based on the power traces measured on an open DPA evaluation board with Xilinx Artix 7. Different distinguishers were used to obtain a key guess:

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

We have evaluated these distinguishers using two different revisions of the board: Table I presents the results when using the DPABoard with a switching power supply; Table II presents the results when using the DPABoard with a linear power supply.

The performance of First derivative distinguisher is much better than the performance of standard CPA (Maximum Pearson correlation coefficient) in case of noisy traces obtained from board with a switching power supply. While in case of First derivative approach we needed just 2,500 power traces to succesfully reveal all 16 bytes of the key, standard CPA 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.

Even in noiseless environment with a linear power supply, our method provides slightly 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 a standard CPA we needed 2,500 traces 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.

B. Sakura-G (Xilinx Spartan 6)

Table III presents the results for Sakura-G board, equipped with two Xilinx Spartan 6 chips and a 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.

C. Evariste III + Altera Cyclone III board

Table IV presents the results for the board with Altera Cyclone III chip and a linear power supply. In this case, first derivative distinguishers and standard CPA are comparable

Table III
NUMBER OF CORRECTLY GUESSED BYTES OF THE KEY, SAKURA-G (XILINX SPARTAN 6 WITH A LINEAR 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)	2	3	6	12	16	16	16	16	16	16
Laplacian of Gaussian (d=25, σ=12)	1	2	5	11	16	16	16	16	16	16

Table IV
NUMBER OF CORRECTLY GUESSED BYTES OF THE KEY, ALTERA CYCLONE III WITH A LINEAR 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)	2	3	4	11	16	16	16	16	16	16
Laplacian of Gaussian (d=25, σ=10)	0	1	1	2	2	2	6	8	11	11

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.

V. CONCLUSION

We have presented a new 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 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 traces obtained in noisy environment, such as that made by the switching power supplies. Using our method, approximately 2,500 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 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.

ACKNOWLEDGMENT

This research has been partially supported by 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] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Investigations of power analysis attacks on smartcards." *Smartcard*, vol. 99, pp. 151–161, 1999.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.