Improving Side-Channel Analysis with Optimal Linear Transforms

David Oswald and Christof Paar

Horst Görtz Institute for IT Security
Ruhr-University Bochum, Germany
{david.oswald,christof.paar}@rub.de

Abstract. Pre-processing techniques are widely used to increase the success rate of side-channel analysis when attacking (protected) implementations of cryptographic algorithms. However, as of today, the according steps are usually chosen heuristically. In this paper, we present an analytical expression for the correlation coefficient after applying a linear transform to the side-channel traces. Doing so, we are able to precisely quantify the influence of a linear filter on the result of a correlation power analysis. On this basis, we demonstrate the use of optimisation algorithms to efficiently and methodically derive "optimal" filter coefficients in the sense that they maximise a given definition for the distinguishability of the correct key candidate. We verify the effectiveness of our methods by analysing both simulated and real-world traces for a hardware implementation of the AES.

Keywords: side-channel analysis, linear filtering, countermeasures, pre-processing, attacks on protected implementations, DPA contest v2.

1 Introduction

The use of Digital Signal Processing (DSP) to facilitate attacks based on Side-Channel Analysis (SCA) or to reduce the number of needed measurements (*traces*) has been demonstrated to be effective in numerous publications (cf. Sect. 1.1). Methods such as Differential Power Analysis (DPA) [12] or Correlation Power Analysis (CPA) [5] often benefit from prior signal processing, e.g., Finite Impulse Response (FIR) filtering. Yet, the precise effect of the pre-processing steps on the success rate of SCA has, to our knowledge, not been precisely quantified. In this paper, we utilise analytical properties of the correlation coefficient in order to derive a more systematic approach for optimising the – so far mostly heuristically selected – pre-processing parameters. From a designer's point of view, our method helps to objectively estimate the amount of leakage an adversary might extract by means of filtering.

1.1 Related Work

One of the first examples of DSP being applied to SCA can be found in [14]: the authors mention the use of a matched filter to increase the Signal-to-Noise

S. Mangard (Ed.): CARDIS 2012, LNCS 7771, pp. 219–233, 2013.

Ratio (SNR) and thus the height of a DPA peak. In [6], Clavier et al. propose to perform comb filtering, i.e., average measurement samples from multiple clock cycles in order to increase the success rate of a DPA in the presence of random process interrupts.

Especially for practical attacks on cryptographic devices, DSP pre-processing is often mandatory, for instance because of uncorrelated noise due to non-cryptographic parts of an Integrated Circuit (IC). In [2], digital filtering helped to isolate the frequency components containing the side-channel leakage of a cryptographic co-processor. Similarly, the attacks on the bitstream encryption mechanism of Xilinx Field Programmable Gate Arrays (FPGAs) required the removal of an interfering signal [15].

For SCA utilising the electro-magnetic (EM) emanation of a cryptographic device, digital filters have been applied to isolate the frequencies containing the side-channel leakage [1]. In the context of cryptographic Radio Frequency Identification (RFID) devices, DSP pre-processing steps have been shown to be necessary [18]. Accordingly, the real-world attacks on the Mifare DESFire MF3ICD40 RFID smartcard described in [11,17] involve several filter operations.

Yet, there is almost no systematic research how DSP operations such as filtering affect the outcome of SCA. In [3], the authors propose an approach to automatically determine appropriate bandpass filters, using CPA as a block algorithm that is repeatedly executed for different choices for the filter coefficients. In general, however, filtering is seen as a completely separate pre-processing step, and the parameters are usually chosen manually.

1.2 Contribution of This Paper

In this paper, we intend to improve on the current approach for devising suitable DSP operations for SCA. More precisely, we examine the effect of linear transforms (which cover amongst others linear filters) on the result of a CPA. The remainder of this paper is organised as follows: in Sect. 2, we briefly review CPA and linear filtering. Then, we introduce a matrix notation that provides a closed form for the correlation coefficient after a linear transform in Sect. 2.1. On this basis, we propose the use of numerical optimisation to determine "good" filter coefficients in Sect. 3. In Sect. 4 and Sect. 5, we compare our approach to normal CPA and to a CPA in the frequency domain, using simulated and real-world measurements (provided in the second DPA contest [7]), respectively. Finally, we conclude in Sect. 6.

2 CPA and Linear Transforms

In the following, we assume the usual setting of SCA: the adversary sends freely chosen input data to a Device Under Test (DUT) (that performs a cryptographic operation on this data) and obtains the corresponding output. The computation done by the DUT involves some secret information (in the following referred to as a key) $k^{\text{dut}} \in K$ (with K the set of all possible keys) that the adversary aims to obtain by means of SCA.

Notation. The process of performing a CPA can be divided into two steps: in the measurement phase, the adversary has physical access to the DUT and records some side-channel signal (e.g., the power consumption or the electromagnetical emanation during the cryptographic computation) that is related to the processed data. This step is repeated N times with varying input data M_i, yielding N time-discrete waveforms $x_i\left(t\right)$ with T points each. In the evaluation phase, the key is recovered by fixing a (small) subset $\mathcal{K}_{cand} \subseteq \mathcal{K}$ and considering all key candidates $k \in \mathcal{K}_{cand}$: for each $k \in \mathcal{K}_{cand}$ and for each $i \in \{0, \ldots, N-1\}$, a hypothesis $V_{k,\, i}$ on the value of some intermediate is computed. Using a power model f, this value is then mapped to $h_{k,\, i} = f\left(V_{k,\, i}\right)$ to describe the physical process that causes the side-channel leakage. In practice, for DUTs such as FPGAs or Microcontrollers (μCs), the power model is often either the Hamming Weight (HW) or Hamming Distance (HD) model [13].

$h_{k,\, i}$ and $x_i\left(t\right)$ are treated as observations of discrete random variables. In order to detect the dependency between $h_{k,\, i}$ and $x_i\left(t\right)$, the *correlation coefficient* $\rho_k\left(t\right)$ (for each point in time $t \in \{0, \ldots, T-1\}$ and each key candidate $k \in \mathcal{K}_{cand}$) is given as $\rho_k\left(t\right) = {}^{cov(x(t),\, h_k)}\!/\!\sqrt{var(x(t))var(h_k)}$ with $var\left(\cdot\right)$ indicating the sample variance and $cov\left(\cdot,\, \cdot\right)$ the sample covariance according to the standard definitions [24]. The key candidate $\hat{k}$ with the maximum correlation $\hat{k} = \arg\max_{k,\, t} \rho_k\left(t\right)$ is assumed to be the secret key k^{dut} used by the DUT.

Linear Filters. As mentioned in Sect. 1.1, a CPA in the time domain is often preceded by a linear FIR filter: for example, a bandpass or bandstop filter may be used to isolate or remove certain frequencies present in a side-channel signal [2]. Integrating over multiple clock cycles can be interpreted as a comb filter [6].

An $S-1$-th order FIR filter is defined by S coefficients $a_i \in \mathbb{R}$, $i = 0 \ldots S-1$. The response $y\left(t\right)$ of the filter to the input signal $x\left(t\right)$ is computed as a (sliding) weighted sum of the points of the input signal, i.e., $y\left(t\right) = \sum_{i=0}^{S-1} a_i x\left(t-i\right)$.

In the following Sect. 2.1, we show how to compute the correlation coefficient between a prediction h_k and an arbitrary weighted sum like an FIR filter, given the "raw" correlation for each point in time and the covariance matrix of the input signal. To this end, we apply a matrix notation according to [10].

2.1 Matrix Notation

To improve the readability, we drop the index k for the key candidate in this section. All involved quantities are represented as vectors or matrices. A trace $x_i\left(t\right)$ is hence denoted as $T \times 1$ vector $\boldsymbol{x_i}$. $\Sigma_{\boldsymbol{xx}}$ is the $T \times T$ sample covariance matrix over all traces according to the standard definition.

For the purposes of this paper, the prediction is a scalar h_i, i.e., a 1×1 vector. Note that this restriction is not mandatory: the prediction could also be extended to a $P \times 1$ vector $\boldsymbol{h_i}$, for example to handle multiple bits of one prediction separately. Again, $\Sigma_{\boldsymbol{hh}}$ is the $P \times P$ covariance matrix. For the scalar case $P = 1$, this reduces to the usual variance.

Finally, $\Sigma_{\boldsymbol{xh}}$ is the $T \times P$ covariance matrix between $\boldsymbol{x}$ and $\boldsymbol{h}$. For $P = 1$, this corresponds to the covariance term in the denominator of the traditional formula for the correlation coefficient. Then, given a $T \times 1$ weight vector $\boldsymbol{a}$ and a $P \times 1$ weight vector $\boldsymbol{b}$, a *closed form* for the correlation coefficient between the dot products $\boldsymbol{a} \cdot \boldsymbol{x}_i$ and $\boldsymbol{b} \cdot \boldsymbol{h}_i$ is given by Equation 1 [10].

$$\rho_{\boldsymbol{xh}}\left(\boldsymbol{a},\, \boldsymbol{b}\right) = \frac{\boldsymbol{a}^T \cdot \Sigma_{\boldsymbol{xh}} \cdot \boldsymbol{b}}{\sqrt{\boldsymbol{a}^T \cdot \Sigma_{\boldsymbol{xx}} \cdot \boldsymbol{a}}\sqrt{\boldsymbol{b}^T \cdot \Sigma_{\boldsymbol{hh}} \cdot \boldsymbol{b}}} \tag{1}$$

This representation is fully equivalent to performing the dot products first (as a pre-processing step) and then computing the correlation coefficient on the pre-processed data. In particular, $\boldsymbol{a}$ can be seen as the coefficients of an FIR filter that is applied to each trace separately. Similarly, $\boldsymbol{b}$ corresponds to an arbitrary weighted sum of e.g. several bits of a predicted intermediate. As stated above, for the purposes of this work, we assume a scalar prediction h_i and hence set $P = 1$ and $b = 1$ for the remainder of this paper. As a side note, in order to obtain the unfiltered correlation coefficient at time index $t = 0, 1, \ldots$, only the t'th entry of $\boldsymbol{a}$ has to be set to a non-zero value, i.e., $\boldsymbol{a} = \left(1\,0\,0\,\ldots\,0\right),\, \left(0\,1\,0\,\ldots,0\right),$ and so on.

Note that Equation 1 can be naturally extended to incorporate a *transform matrix* rather than a vector. In this case, $\boldsymbol{a}$ becomes a $T \times S$ matrix A formed by S different column vectors $\boldsymbol{a}_s$, $s \in 0, \ldots\, S-1$. The $S \times 1$ correlation coefficient vector $\boldsymbol{\rho}_{\boldsymbol{xh}}\left(A,\, \boldsymbol{b}\right)$ is then given by evaluating Equation 1 for all $\boldsymbol{a}_s$ and concatenating the results. This form covers (amongst other linear transforms) any FIR filter: A consists of the filter coefficient vector that is shifted by s positions for row s, that is, A is a Toeplitz matrix [21].

Computational Complexity. The form of Equation 1 is useful when the correlation coefficient is to be computed for fixed $\boldsymbol{x}_i$ and h_i but for (many) different $\boldsymbol{a}$: first computing $\boldsymbol{a} \cdot \boldsymbol{x}_i$ for each trace and then evaluating the traditional formula for the correlation coefficient has a complexity of $\mathcal{O}\left(N\,T\right)$. For L different choices $\boldsymbol{a}^l$, $l = 0 \ldots L-1$, the overall effort is thus $\mathcal{O}\left(L\,N\,T\right)$. In contrast, to evaluate Equation 1, the computation of the covariance matrices needs $\mathcal{O}\left(N\,\left(T^2 + T\right)\right) = \mathcal{O}\left(N\,T^2\right)$ operations. The post-processing to obtain the desired correlation coefficients for all $\boldsymbol{a}^l$ is then $\mathcal{O}\left(L\,T^2\right)$ (which is independent of N). Hence, the total complexity is $\mathcal{O}\left(N\,T^2 + L\,T^2\right)$.

Complexity Reduction in Special Cases. The main drawback of Equation 1 is that it requires the covariance matrix $\Sigma_{\boldsymbol{xx}}$. For large T, the estimation of this matrix is problematic due to issues with the computational complexity. Thus, the application of the closed form of the correlation may become infeasible. Incidentally, the statistical efficiency is not an issue in this case — Equation 1 does not involve the inverse of $\Sigma_{\boldsymbol{xx}}$ and is fully valid for small sample size N.

Still, for a filter of order $S-1$, $\boldsymbol{a}$ only has S consecutive non-zero coefficients. Hence, in this case, it is sufficient to estimate $\Sigma_{\boldsymbol{xx}}$ as a band matrix with a bandwidth of S. The computational complexity is then reduced to $\mathcal{O}\left(S\,T\right)$, i.e., linear with respect to the length of the traces. Finally, for the optimisation approach proposed in the following Sect. 3, $\Sigma_{\boldsymbol{xx}}$ can be factored out when determining $\boldsymbol{a}$. Hence, if estimating $\Sigma_{\boldsymbol{xx}}$ becomes prohibiting for large T, one may still utilise the traditional approach and compute the dot products $\boldsymbol{a} \cdot \boldsymbol{x}_i$ before the CPA once the optimal $\boldsymbol{a}$ has been found.

3 Optimal Linear Transforms for CPA

Given the closed-form expression for the transformed correlation coefficient of Equation 1, we propose a method to find "optimal" filter coefficients $\boldsymbol{a}$ in the sense that the filter maximises the distinguishability of the correct key candidate.

To achieve this goal, we regard Equation 1 as a multivariate function in $\boldsymbol{a}$ and employ standard numerical optimisation algorithms. As we aim to maximise the distinguishability rather than the correlation itself, a suitable optimisation criterion has to be defined. We assume a (semi-)profiled scenario in which an adversary possesses an instance of the DUT for which he knows the secret key. Note that the adversary is not necessarily able to change the key — only the knowledge of the correct key is required for the optimisation of the filter coefficients. In contrast to template attacks, which have been shown to be highly sensitive to process variations [19], we expect the filter coefficients to be less sensitive in this regard. This conjecture is based on the fact that a filter modifies the frequency spectrum (which should be less device-dependent than e.g. the signal amplitude), while the actual key recovery is still carried out by a (more robust) differential technique like CPA.

In our experiments, directly maximising Equation 1 gave rise to overfitting of the coefficients $\boldsymbol{a}$. As a result, the correlation is maximised for one specific problem instance (i.e., fixed input data, key, and traces), however, if any parameter changes, the determined coefficients no longer lead to the desired result. Hence, we devised the criterion given in Equation 2. The goal is to maximise the ratio between the absolute value of the correlation coefficient for the correct key k^{dut} and the average over the absolute value of the correlation coefficients for incorrect key candidates $k_{wrong} \in \mathcal{K}_{optim}$, $\mathcal{K}_{optim} = \mathcal{K}_{cand} \setminus \left\{k^{\text{dut}}\right\}$.

$$f_{objective}\left(\boldsymbol{a}\right) = \frac{\left|\rho_{\boldsymbol{x}h_{k^{\text{dut}}}}\left(\boldsymbol{a}\right)\right|}{{}^{1}\!/\!_{\left|\mathcal{K}_{optim}\right|}\left(\sum_{k \in \mathcal{K}_{optim}} \left|\rho_{\boldsymbol{x}h_k}\left(\boldsymbol{a}\right)\right|\right)} \tag{2}$$

Note that in Equation 2, every $\rho_{\boldsymbol{x}h}$ both in the numerator and denominator contains a positive factor of ${}^{1}\!/\!\sqrt{\boldsymbol{a}^T \cdot \Sigma_{\boldsymbol{xx}} \cdot \boldsymbol{a}}$ (independent of h_k) which can be cancelled. Equation 2 thus takes the form of Equation 3 (whereas the factor ${}^{1}\!/\!_{\left|\mathcal{K}_{optim}\right|}$ was left out).

$$f_{objective}\left(\boldsymbol{a}\right) = \frac{\left|{}^{1}\!/\!\sqrt{\Sigma_{h_{k^{\text{dut}}}h_{k^{\text{dut}}}}} \cdot \boldsymbol{a}^T \cdot \Sigma_{\boldsymbol{x}h_{k^{\text{dut}}}}\right|}{\sum_{k \in \mathcal{K}_{optim}} \left|{}^{1}\!/\!\sqrt{\Sigma_{h_k h_k}} \cdot \boldsymbol{a}^T \cdot \Sigma_{\boldsymbol{x}h_k}\right|} \tag{3}$$

This eliminates the computationally most expensive part of Equation 1, namely the vector-matrix product with complexity $\mathcal{O}\left(T^2\right)$. Moreover – at least in the profiling step – the covariance matrix $\Sigma_{\boldsymbol{xx}}$ is not needed at all. Hence, as mentioned in Sect. 2.1, the optimisation of the weight coefficients can be carried out even with long traces for which computational issues make the estimation of the sample covariance matrix difficult or impossible. To numerically find an optimum of $f_{objective}$, we employ the function `fminunc` provided by the MATLAB optimization toolbox [23]. This function *minimises* a given objective function. In our case, we thus search for a minimum of $-f_{objective}$ (which is equivalent to a maximum of $f_{objective}$).

3.1 Relation to Other Techniques

Principal Component Analysis. The method of Principal Component Analysis (PCA) [22] transforms signals to a new (lower-dimensional) representation. Recently, Batina et al. proposed to use PCA as a pre-processing step for a CPA [4]. Their idea is based on the observation that a leakage signal and unrelated noise are often mapped to different principal components. PCA is a linear transform, i.e., a trace $\boldsymbol{x}_i$ is projected to the new representation using the vector-matrix product $\boldsymbol{y} = U^T \cdot \boldsymbol{x}_i$ with U the matrix of the (retained) eigenvectors of the covariance matrix $\Sigma_{\boldsymbol{xx}}$. Thus, as mentioned in Sect. 2.1, one point of the projected trace $\boldsymbol{y}$ is given as scalar product between $\boldsymbol{x}_i$ and one row of U^T. The rows of U^T can therefore be regarded as different choices for the weight vector.

Canonical Correlation Analysis. In contrast to PCA which picks principal components with maximum variance, Canonical Correlation Analysis (CCA) [10] finds a weight vector that maximises the correlation coefficient. Performing an eigenvector decomposition of the covariance matrix, CCA finds a vector $\boldsymbol{a}$ that maximises Equation 1. However, as mentioned in Sect. 3, in our experiments this lead to overfitting and resulted in non-applicable weight vectors.

SCA in the Frequency Domain. Transforming traces to the frequency domain and discarding the phase component has been shown to be beneficial for SCA [8,18]. This pre-processing step, also known as Differential Frequency Analysis (DFA), is both applicable to overcome misalignment in the traces and to spectrally isolate the leakage component. Note that the phase component is removed by taking the absolute value of the transformed traces, i.e., $|\mathrm{DFT}\{\boldsymbol{x}_i\}|$. Due to the absolute value operator, the transform is no longer linear, and hence cannot be described in terms of Equation 1 with a suitable $\boldsymbol{a}$. In Sect. 4 and Sect. 5, we provide a comparision of our proposed technique to DFA.

It should be taken into account that computing the Discrete Fourier Transform (DFT) over the complete trace is only suitable in special cases. In practice, a trace is usually split into windows of a given length which are processed separately [17]. As of today, determining the optimal window length is a somewhat heuristic process that either involves (educated) guessing or optimisation by testing many choices for the parameter.

Note that our proposed method can be combined with the frequency domain transformation. The weight vector is then applied to the transformed traces $|\mathrm{DFT}\{\boldsymbol{x}_i\}|$ and optimised according to Sect. 3. In cases where the leakage is distributed over multiple frequency bins, this approach can combine and thus presumably better utilise the overall side-channel information. Therefore, we also included this approach in our simulation and practical results in Sect. 4 and Sect. 5.

4 Simulation Results

In order to evaluate the effect of the optimised weight coefficients, we generated simulated traces for a 128-bit implementation of the Advanced Encryption Standard (AES) in MATLAB. The main purpose of this section is to demonstrate the basic effectiveness of the proposed approach. We do not aim to comprehensively examine every conceivable scenario, hence, the choice of the simulation parameters may appear somewhat arbitrary.

In our simulation, the clock frequency was set to $33.\overline{3}$ MHz, with the trace being sampled at 1 GHz. A clock cycle thus contains 30 samples. The i'th simulated trace for the clock cycle c is then generated as the sum of a "clock" signal multiplied by a leakage s_i^c and normally distributed noise as $\boldsymbol{x}_i^c = (1 + \sigma_{signal} \cdot s_i^c)\ \boldsymbol{t} + \boldsymbol{\mathcal{N}}(0, \sigma_{noise})$

We used $\sigma_{signal}^2 = \sigma_{noise}^2 = {}^1\!/_{1000}$. $\boldsymbol{t}$ was set to a rectangular pulse, that is, $\boldsymbol{t} = (1, 1, \ldots, 1, 0, 0, \ldots, 0)$, with the first seven entries set to 1 (corresponding to ${}^1\!/_4$ of the full cycle) and the remaining 23 entries set to 0. To form the final simulated trace $\boldsymbol{x}_i$, we concatenated four cycles $\boldsymbol{x}_i^c$. The leakage in the first cycle s_i^0 was generated as the HW of the 128-bit AES state after the initial key addition and SubBytes operation. In the remaining three cycles, $s_i^{1/2/3}$ was calculated as the HW of a uniformly distributed random 128-bit value.

Band-Limited Noise. To simulate the effect of a band-limited noise source, we added an additional noise term $\boldsymbol{\mathcal{N}}_{band}$ to the simulated trace. For our experiments, we selected a noise bandwidth of ±1 MHz around 24 MHz, i.e., the spectrum of $\boldsymbol{\mathcal{N}}_{band}$ is "white" between 23 and 25 MHz and zero otherwise. For a range of noise powers of $\boldsymbol{\mathcal{N}}_{band}$, we then performed (1) a time domain CPA, (2) a CPA on the frequency domain representation of the traces (cf. Sect. 3.1), (3) a time domain, and (4) a frequency domain CPA using optimised filter coefficients $\boldsymbol{a}$ as described in Sect. 3. For the profiling and the attack, we used different keys and different input data.

Fig. 1 depicts the respective (maximum) correlation for the cases (1), (2), (3), and (4) for a noise power (i.e., the average standard deviations $\sigma_{bandlimited}$) of 1. The average signal power of a trace was $\sigma_{trace} = 0.56$, i.e., the given $\sigma_{bandlimited}$ correspond to a "Trace-to-Noise Ratio" (TNR) of approximately 0.5. Because the simulated traces already contain white noise, we avoid the term SNR here.

As evident in Fig. 1, the CPA using optimised filter coefficients (3) outperforms the normal CPA (1) and the frequency domain CPA (2) in the presence of

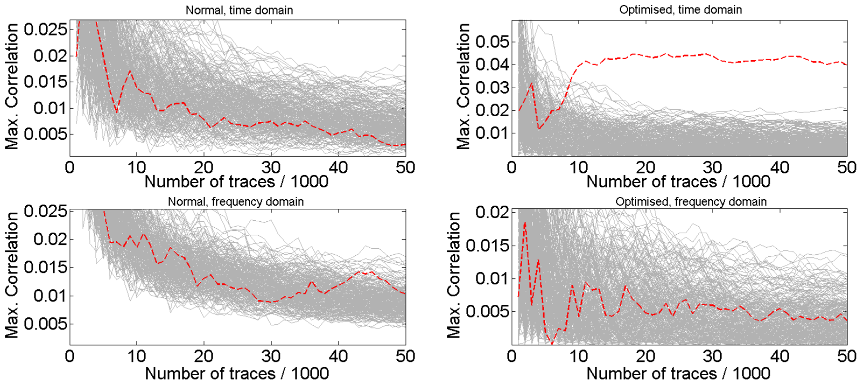


Fig. 1. Maximum correlation for band-limited noise, $\sigma_{bandlimited} = 1$. Top left: time domain (1), top right: time domain, optimised (3), bottom left: frequency domain (2), bottom right: frequency domain, optimised (4). Correct key candidate: dashed, red.

band-limited noise. Table 1 summarises the results, giving the absolute value of the correlation coefficient for the correct key after 50,000 traces (Table 1a) and the ratio between the correlation for the correct candidate and the maximum correlation for the wrong candidates (Table 1b). If this ratio is less than 1, the correct key can no longer be distinguished from the wrong candidates, i.e., the attack does not succeed.

Table 1. Comparision of evaluation methods (1) - (4) for simulated traces with band-limited noise. Best values in bold font.

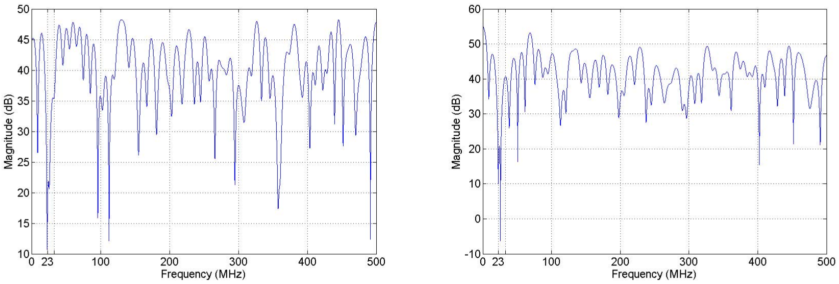
TNR	(1)	(2)	(3)	(4)
∞	**0.123**	0.09	0.051	0.017
1.7	0.009	0.019	**0.049**	0.014
1	0.005	0.012	**0.049**	0.007
0.5	0.003	0.01	**0.04**	0.004

(a) Correlation using 50k traces

TNR	(1)	(2)	(3)	(4)
∞	**3.73**	3.46	3	1.21
1.7	0.5	1	**2.88**	1.17
1	0.28	0.63	**2.72**	0.44
0.5	0.17	0.59	**1.9**	0.26

(b) Ratio between correct and maximum wrong candidate (50k traces)

With increasing $\sigma_{bandlimited}$ (i.e., decreasing TNR), the correlation coefficient for the correct key candidate is very close to or even below the correlations for the wrong key candidates using method (1), (2), or (4) after 50,000 traces. In contrast, the correlation for the correct key obtained with method (3) clearly exceeds the correlation for the wrong candidates after less than 5,000 traces in all cases. Computing the frequency response corresponding to the optimised coefficients, it turns out that the range from 23 to 25 MHz is attenuated, while the filter's transfer function is rather flat in the region of the clock frequency. The corresponding plot of the frequency response is given in Fig. 2a.



(a) Noise ($\sigma_{bandlimited} = 1$) (b) Noise + randomised timing (4 cycles)

Fig. 2. Magnitude frequency response of the optimised filter coefficients for the simulated traces

Interestingly, if no additional noise is present, method (3) provides worse distinguishability of the correct key than methods (1) and (2). In this case, the optimisation algorithm appears to overfit the weight coefficients – the coefficients then yield optimal distinguishability based on the data used in the profiling phase, but do not produce the desired effect in general. Thus, in the attack phase with a different set of traces, the distinguishabilty is reduced and not – as intended — increased. This problem could presumably be mitigated by techniques used in global optimisation, e.g., running the optimisation algorithm several times using different initial values and different subsets of the profiling data. For the purposes of this paper, we did not look further into this issue and leave it for future work.

Timing Randomisation. A randomisation of the algorithmic timing was realised by shuffling the four clock cycles for each trace, that is, by randomly selecting a uniformly distributed position for the clock cycle that corresponds to the actual AES state. For this case, we applied the same evaluation methods as above. We also combined the timing randomisation with the band-limited noise source, again considering the same range of noise powers as for the non-randomised traces.

The result for $\sigma_{bandlimited} = 1$, i.e., a TNR of approximately 0.5, is exemplarily depicted in Fig. 3. Table 2 subsumes the results like in Table 1, giving the maximum correlation and the ratio between the correlation for the correct and the highest wrong candidate for different TNRs. While the normal CPA and the frequency domain CPA fail to clearly distinguish the correct key candidate from the wrong ones after 50,000 traces, the optimisation approach determines filter coefficients that allow to extract the correct candidate after less than 5,000 traces. The according frequency response (Fig. 2b) again exhibits a band-stop characteristic eliminating the band-limited noise. In the time domain, the filter coefficients additionally resemble a comb filter, i.e., realise the averaging over multiple clock cycles [6].

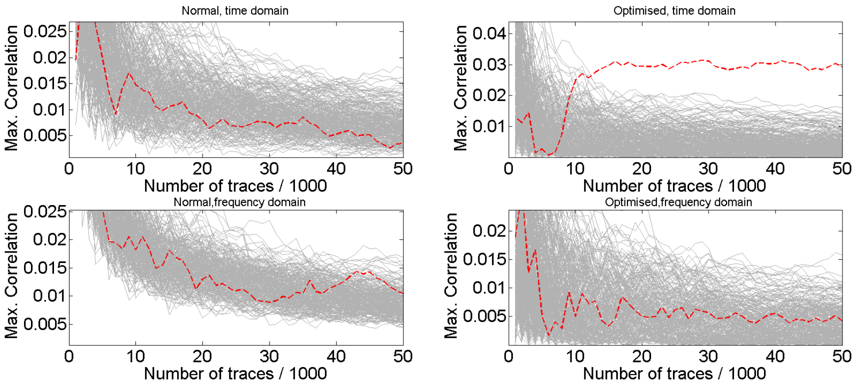


Fig. 3. Maximum correlation for jitter (4 cycles) and band-limited noise, $\sigma_{bandlimited} = 1$. Top left: time domain (1), top right: time domain, optimised (3), bottom left: frequency domain (2), bottom right: frequency domain, optimised (4). Correct key candidate: dashed, red.

Table 2. Comparision of evaluation methods (1) - (4) for simulated traces with band-limited noise and timing randomisation (4 cycles). Best values in bold font.

TNR	(1)	(2)	(3)	(4)
∞	0.038	**0.089**	0.04	0.02
1.7	0.005	0.018	**0.04**	0.014
1	0.004	0.011	**0.038**	0.007
0.5	0.003	0.01	**0.029**	0.004

(a) Correlation using 50k traces

TNR	(1)	(2)	(3)	(4)
∞	1.9	**3.42**	2.5	1.33
1.7	0.28	1	**2.22**	1.08
0.5	0.22	0.58	**2.11**	0.54
1	0.17	0.59	**1.81**	0.25

(b) Ratio between correct and maximum wrong candidate (50k traces)

5 Practical Results

In order to evaluate the efficiency of our findings in a real-world setting, we applied our methods to the traces provided in the second DPA contest [7]. The traces were recorded for a hardware implementation of the AES on the Sasebo GII [16] at a sample rate of $f_s = 5$ GHz. We focused on the last round of the encryption process and accordingly only used the respective part from time point 2300 to 2700 of the 3253-point original traces.

For the profiling purposes, we used 15,000 raw traces of the "public database" (`DPA_contest2_public_base_diff_vcc_a128_2009_12_23`) belonging to the encryption with the AES key $k_{profiling} =$ `0x37d0d724d00a1248db0fead349f1c09b`. For the attack phase, i.e., to evaluate the effect of the optimised filter coefficients, we used 15,000 traces for $k_{attack} =$ `0x0000000000000003243f6a8885a308d3`. These keys lead to different subkeys for the first S-Box in the final round (`0xdc` for $k_{profiling}$, `0x53` for k_{attack}).

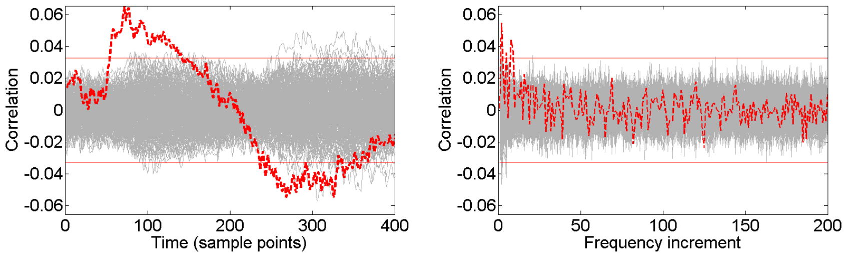


Fig. 4. Correlation coefficients (DPA contest v2 AES) for the first byte after 15,000 traces, left: time domain, right: frequency domain. Correct key candidate: dashed, red.

As a first step, we performed a standard CPA targeting the (bytewise) Hamming distance between the input of the last SubBytes operation and the encryption result, following the reference attack of the DPA contest. As depicted in Fig. 4, the highest correlation coefficient clearly occurs for the correct key candidate after 15,000 traces, with a magnitude of 0.064 in the time domain and 0.055 in the frequency domain, respectively. However, significant "ghost peaks" occur at the end of the trace (around point 300).

At this point, we want to emphasise that we primarily use the DPA contest traces to demonstrate the general effectivity of our approach in a practical setting. Hence, we did not employ the full range of evaluation metrics provided by the contest. We applied the same evaluation methods as in Sect. 4, that is, CPA (1), frequency domain CPA (2), CPA with optimised coefficients (3), and frequency domain CPA with optimised coefficients (4).

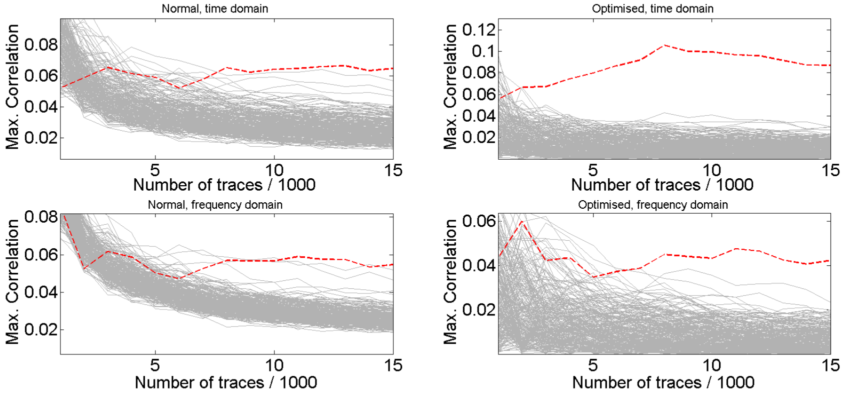
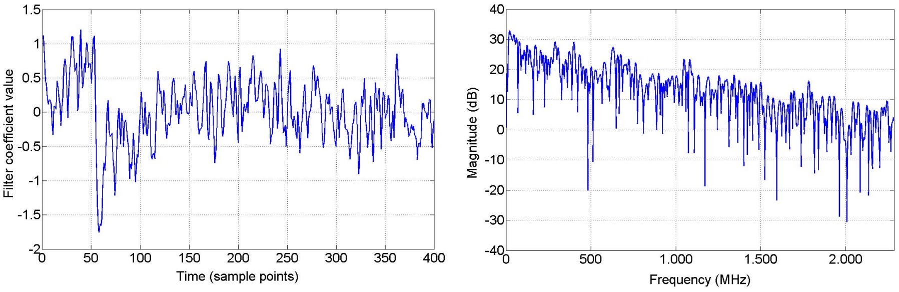


Fig. 5. Maximum correlation coefficients (DPA contest v2 AES) for the first byte. Top left: time domain (1), top right: time domain, optimised (3), bottom left: frequency domain (2), bottom right: frequency domain, optimised (4). Correct key candidate: dashed, red.



(a) Time domain

(b) Frequency response for (a)

Fig. 6. Optimised filter coefficients for the DPA contest v2 traces

As evident in Fig. 5, the optimised coefficients decrease the number of required traces both for the time and the frequency domain CPA: for the normal CPA, the correct key candidate yields the highest correlation, however, the ratio with the second highest is rather small, i.e., $^{0.064}/_{0.057} = 1.12$ in the time domain and $^{0.055}/_{0.052} = 1.06$ in the frequency domain. In contrast, with the optimised filter coefficients, these ratios are increased to $^{0.087}/_{0.03} = 2.9$ and $^{0.042}/_{0.023} = 1.83$, respectively. Accordingly, the (approximate) minimum number of traces needed to distinguish the correct and the wrong key candidate is reduced from 8,000 to 3,000 in the time domain and 11,000 to 8,000 in the frequency domain.

The optimised coefficients reduce the influence of the "ghost peaks" mentioned above on the results of the CPA, i.e., improve the distinguishability of the correct key candidate. As evident in Fig. 6a, the optimised coefficients obtained with method (3) put the highest weight on the maximum of the leakage at around time point 50, followed by decaying weight according to the shape of the correlation depicted in Fig. 4. The according frequency response (Fig. 6b) shows a lowpass characteristic in general. However, certain frequencies are selectively attenuated, for example, narrow regions around 70 MHz, 160 MHz, 227 MHz, 327 MHz, 388 MHz, 422 MHz, and 480 Mhz.

We experimentally verified that these results obtained for the first byte equivalently hold for the other bytes. Table 3 gives the ratio between the correlation for the correct and the highest wrong candidate after 15,000 traces for a normal CPA (1) and a CPA with optimised coefficients (3) in the time domain. For all

Table 3. Ratio between correct and maximum wrong candidate for normal (1) and optimized CPA (3) in the time domain after 15,000 DPA contest v2 traces

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CPA	1.12	1.84	1.47	1.67	1.02	1.77	1.5	1.45	1.68	2.17	1.24	1.83	1.63	1.68	1.53	1.81
opt. CPA	2.9	2.74	2.48	3.83	1.71	3.88	3.12	3.15	3.44	3.79	2.16	3.31	2.49	3.24	2.54	4.44

bytes, the optimised coefficients lead to a higher distinguishability and allow for extracting the key with less traces.

6 Summary

We presented a closed form to perform CPA on traces under a linear transform. In contrast to traditional approaches, our method does not require to re-compute the CPA for each particular choice of the transform parameters. Thus, in cases where many different parameters are to be tested, our method allows for a substantially faster evaluation. Consequently, we derived an optimisation criterion allowing to find an “optimal” transform in the sense that it maximises the distinguishability of the correct key candidate. Using both simulated and real-world traces, we demonstrated that this technique performs better than traditional methods and offers a systematic way to derive linear filters for SCA. Especially when designing countermeasures against SCA, our method allows to give a more comprehensive (and objective) assessment regarding the effectiveness of protection mechanisms.

Future Work. Our work offers several starting points for further research: first of all, the employed numerical optimisation algorithm was used “out-of-the-box”. We believe that an algorithm adapted to the specific requirements of our method may lead to better results and avoid the problem of overfitting. Besides, the proposed optimisation criterion could be replaced, utilising a different metric for the distinguishability of the correct key candidates. It would also be interesting to investigate whether an analytical solution for the present or a different suitable optimisation criterion can be computed efficiently. In this regard, the applicability of statistical methods like CCA in a side-channel context would deserve some attention.

We limited our experiments to the weight coefficients applied to the traces. However, equivalently, the prediction could also be subject to a linear transform. This essentially corresponds to finding a suitable model for the contribution of single bits to the overall leakage, i.e., relates to SCA with stochastic models [20]. Finally, we focused on CPA only. However, distinguishers like Mutual Information Analysis (MIA) [9] have been shown to be superior in certain cases. Thus, finding a similar technique to compute the mutual information of transformed traces without re-executing the complete MIA is worth further research.

Acknowledgements. The work described in this paper has been supported in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II and by the German Federal Ministry of Education and Research BMBF (grant 01IS10026A, Project EXSET).

References

1. 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)
2. Barenghi, A., Pelosi, G., Teglia, Y.: Improving First Order Differential Power Attacks through Digital Signal Processing. In: Proceedings of the 3rd International Conference on Security of Information and Networks, SIN 2010, pp. 124–133. ACM, New York (2010)
3. Barenghi, A., Pelosi, G., Teglia, Y.: Information Leakage Discovery Techniques to Enhance Secure Chip Design. In: Ardagna, C.A., Zhou, J. (eds.) WISTP 2011. LNCS, vol. 6633, pp. 128–143. Springer, Heidelberg (2011)
4. Batina, L., Hogenboom, J., van Woudenberg, J.G.J.: Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 383–397. Springer, Heidelberg (2012)
5. Brier, E., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
6. Clavier, C., Coron, J.-S., Dabbous, N.: Differential Power Analysis in the Presence of Hardware Countermeasures. In: Koç, Ç.K., Paar, C. (eds.) CHES 2000. LNCS, vol. 1965, pp. 252–263. Springer, Heidelberg (2000)
7. COMELEC department, Télécom ParisTech. DPA Contest v2. Website, `http://www.dpacontest.org/v2/index.php`
8. Gebotys, C.H., Ho, S., Tiu, C.C.: EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 250–264. Springer, Heidelberg (2005)
9. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual Information Analysis – A Generic Side-Channel Distinguisher. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 426–442. Springer, Heidelberg (2008)
10. Hardoon, D.R., Szedmak, S., Shawe-Taylor, J.: Canonical Correlation Analysis: An Overview with Application to Learning Methods (May 2003)
11. Kasper, T., Oswald, D., Paar, C.: Side-Channel Analysis of Cryptographic RFIDs with Analog Demodulation. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 61–77. Springer, Heidelberg (2012)
12. 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)
13. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer, Secaucus (2007)
14. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Investigations of Power Analysis Attacks on Smartcards. In: USENIX Workshop on Smartcard Technology, pp. 151–162 (1999)
15. Moradi, A., Barenghi, A., Kasper, T., Paar, C.: On the Vulnerability of FPGA Bitstream Encryption against Power Analysis Attacks: Extracting keys from Xilinx Virtex-II FPGAs. In: Chen, Y., Danezis, G., Shmatikov, V. (eds.) ACM Conference on Computer and Communications Security (CCS 2011), pp. 111–124 (2011)
16. National Institute of Advanced Industrial Science and Technology (AIST). Side-channel Attack Standard Evaluation Board SASEBO-GII Specification, 1.01 edition (2009)
17. Oswald, D., Paar, C.: Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 207–222. Springer, Heidelberg (2011)

18. Plos, T., Hutter, M., Feldhofer, M.: Evaluation of Side-Channel Preprocessing Techniques on Cryptographic-Enabled HF and UHF RFID-Tag Prototypes. In: Dominikus, S. (ed.) Workshop on RFID Security — RFIDSEC 2008, pp. 114–127 (2008)
19. Renauld, M., Standaert, F.-X., Veyrat-Charvillon, N., Kamel, D., Flandre, D.: A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 109–128. Springer, Heidelberg (2011)
20. Schindler, W., Lemke, K., Paar, C.: A Stochastic Model for Differential Side Channel Cryptanalysis. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 30–46. Springer, Heidelberg (2005)
21. Smith, J.O.: General LTI Filter Matrix. In: Introduction to Digital Filters with Audio Applications. Center for Computer Research in Music and Acoustics (2007), `http://www.dsprelated.com/dspbooks/filters/`
22. Standaert, F.-X., Archambeau, C.: Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 411–425. Springer, Heidelberg (2008)
23. The MathWorks, Inc. MATLAB R2011b Documentation, Optimization Toolbox, fminunc. Website (Online; accessed February 28, 2012)
24. Weisstein, E.W.: Variance. Mathworld - A Wolfram Web Resource (December 2010), `http://mathworld.wolfram.com/Variance.html`