SCA with Magnitude Squared Coherence

Philippe Maurine, Sébastien Tiran

SCA with Magnitude Squared Coherence

Sebastien Tiran and Philippe Maurine

University of Montpellier / LIRMM
161 Rue Ada
34392 Montpellier France

Abstract. Magnitude Squared Coherence (MSC) is a signal processing tool that indicates how well two time domain signals match one with the other by tracking linear dependencies in their spectral decomposition. Spectral Coherence ANalysis (SCAN) was the first way to use it as a Side-Channel Attack (SCA). This paper introduces two ways of using the Magnitude Squared Coherence in side-channel analyses. The first way is to use it as a distinguisher while the second consists in using it to transform the side-channel traces in a worthwhile manner. Additionally, an algorithm for fast computation of the SCAN is provided.

Keywords: Secure Circuits, Side-Channel Attacks, Frequency Domain, Distinguisher

1 Introduction

Following [7], many side-channel attacks have been proposed in the literature. Most of them directly work with time domain traces, and aim at analysing each time sample independently to retrieve the secret key. However, the power consumption and the electromagnetic (EM) emanations of a cryptographic algorithm, are such that the leakage is spread over many time samples. Consequently, these analyses cannot exploit the leakage in its whole.

By contrast, much less attention has been paid to side-channel analyses performed in the frequency domain that could bring a solution to this problem. To the best knowledge of the authors, Gebotys, Ho and Tiu [5] were the first to propose a differential attack after the application of a Fast Fourier Transform (FFT) to side-channel traces. This work was then extended towards Correlation Power Analysis like attacks (CPA) in [2]. Various similar approaches were described in [9]. These works underlined the advantages of frequency domain analyses against misaligned traces but they didn't focus on the fact that they can also capture more efficiently a leakage that is spread over time.

The Magnitude Squared Coherence being a tool that works in the frequency domain to estimate the similarity between two signals, it can be used in the context of side-channel analysis to retrieve the secret key [4, 13]. In some cases, it can provide better results than the time domain distinguishers. The advantages of MSC are that it can exploit the leakage scattered in time and fully use it by exploiting several harmonics.

This paper aims at showing the efficiency of the MSC as a tool for side-channel analysis. First, it shows that the MSC is a really interesting distinguisher and provides some explanations related to its efficiency. Second, a new way of using the MSC is introduced. It consists in transforming the traces to get a wider source of information before exploitation by statistical means.

The rest of this paper is organized as follow. Section 2 reminds some basics about Magnitude Squared Coherence. Section 3 briefly recalls its first use in the context of side-channel analysis. It also provides additional information such as an efficient coding of the SCAN. Section 4 presents a method to transform the side-channel traces and introduces various solutions to exploit the resulting source of information. In section 5, experimental results of these attacks are shown. Finally, a conclusion is drawn.

2 Magnitude Squared Coherence

The Magnitude Squared Coherence is a signal processing tool that returns real values between 0 and 1 to indicate how well two time domain signals *x(t)* and *y(t)* match one with the other. It provides scores, *MSC(f)*, allowing to estimate their similarity at various frequencies. The result of the Magnitude Squared Coherence at a given frequency, *f*, is obtained by computing :

$$MSC(f) = \frac{|P_{xy}(f)|^2}{P_{xx}(f).P_{yy}(f)}. \tag{1}$$

where P_{xy} is the cross-power spectral density of *x(t)* and *y(t)* and P_{xx}, P_{yy} are the auto-power spectral densities of *x(t)* and *y(t)*, respectively. To calculate the cross-power spectral density, the Welch's average method is typically used [10]. This consists in : dividing the signals in several overlapping segments of the same length, computing the cross-power spectral density between each pair of segments and finally, computing the average.

$$P_{x,y}(f) = \sum_{i=1}^{n} FFT_{x_i}(f).FFT_{y_i}(f) \tag{2}$$

with $\{x_1(t), x_2(t), ..., x_n(t)\}$ and $\{y_1(t), y_2(t), ..., y_n(t)\}$ the segments of *x(t)* and *y(t)*, respectively.

3 Spectral Coherence ANalysis

Typically in side-channel analysis, an adversary has to collect a set of traces, $\{T_1, T_2, ..., T_n\}$, corresponding to the encryption / decryption of n messages, $\{m_1, m_2, ..., m_n\}$. He then sorts these traces according to a given selection function, f_k, that predicts some intermediate values computed during the algorithm execution and that depend on a part of the key.

$$\begin{aligned} f_k : \{0,1\}^q \times \{0,1\}^p &\rightarrow \{0,1\}^w \\ (m_i, k) &\rightarrow f_k(m_i, k) = c_i \end{aligned} \tag{3}$$

If the Hamming Weight (HW) model is chosen, then f_k predicts a word c_i of w bits processed by the cryptographic algorithm according to the value of a clear / ciphered message m_i and according to a key guess $k \in K$. If the Hamming Distance (HD) model is preferred, then f_k rather provides a word c_i indicating which of the w bits have switched during a given clock cycle.

To apply a SCAN as defined in [4, 13], one have to proceed as for a DPA. For each guess on the key and for each of the w bits, the adversary must sort the traces in two subsets depending on the values provided by f_k and compute the means of the two resulting subsets. But instead of computing the difference of the means (DoM), one may first compute the Magnitude Squared Coherence between the two mean traces. Finally, one have to compute the mean, of all obtained *MSC(f)* values in order to fix a score for the considered key guess.

For wrong key guesses, as the traces are not well sorted, the two mean traces are expected to be similar while they should be significantly different for the correct key. This is to say that the correct key is identified by searching the guess with the minimum score. For detailed information about efficient implementation of the SCAN, see Appendix A.

As one may conclude from this brief description, the SCAN is obtained from the DPA simply by replacing the DoM by the MSC distinguisher. This is an intuitive and straightforward way of using the MSC within the context of SCA. However this approach could be far from being optimal. Indeed, the MSC can be used differently. For example, one may use it to transform the side-channel traces containing the leakage as explained below.

4 Transformation of the Leakage

The basic idea of leakage transformation is to construct from the set of available traces, a new set of data on which side-channel attacks are more efficient. The MSC offers the possibility of applying this idea.

4.1 Preprocessing Step

Indeed, given a set of traces, one may compute the *MSC(f)* between each pair of curves. By doing so, one gets several *MSC(f)* for each pair of curves. To exploit fully the leakage, which is scattered on many frequencies, one may then computes the mean of all the *MSC(f)* values to get a score, *Coher*(T_i, T_j), that will represent, in the rest of the paper, the difference of leakage between two traces, namely T_i and T_j.

$$Coher(T_i, T_j) = \frac{1}{nbf} . \sum_{f=fmin}^{fmax} MSC_{T_i,T_j}(f) \tag{4}$$

where T_i and T_j are two time domain signals and *nbf* is the number of harmonics falling in the bandwidth of interest, ie between f_{min} and f_{max}, the cut-off frequencies of the used equipments .

At the end of this leakage transformation step, one obtains for a set of n traces, $n.\frac{n-1}{2}$ *Coher*(T_i, T_j) scalars, that is to say $\frac{n-1}{2}$ times as much informers, the latter being related to the leakage difference of trace pairs. Such a multiplication of data constitutes an interesting advantage while attacking systems in which keys are regularly refreshed. The open questions are then:

- what is the relevant leakage model?
- how to efficiently extract the secret key from the statistical distribution of the *Coher*(T_i, T_j) values?

4.2 Leakage Model

When working in time domain with now classical attacks (DPA, CPA ...), an adversary typically uses the HW and HD models. The basic idea on which the HW model relies is that computations ending by a '1' (Vdd) usually consume more energy than computations ending by a '0' (Gnd). Similarly, the HD model is based on the idea that a state change burns much more energy than a calculus ending by the same result than the preceding one.

Considering that the power consumption or EM emanations are additive quantities, the leakage model we adopted is based on the following idea : the incoherence (coherence) of two traces is an increasing (decreasing) function of the difference of their HW or HD. That is to say: the greater the difference of the Hamming Weights is, the more incoherent (less coherent) the corresponding traces are. This choice, which relies on the shape of the traces rather than on the amplitude of samples, can be improved. However, it has lead to interesting experimental results given in the next sections.

4.3 Specific Selection Function

With such a leakage model, the question is now : how to sort coherence values? In other words, what are the relevant selection functions? Considering (3) and following the same reasoning as for the previous leakage model, we defined the following selection function :

$$\begin{aligned} \Delta f_k : \{0,1\}^w \times \{0,1\}^w &\rightarrow \{0,1\}^w \\ (c_i, c_j) &\rightarrow \Delta f_k(c_i, c_j) = f_k(m_i, k) \oplus f_k(m_j, k) = \Delta c_{i,j} \end{aligned} \tag{5}$$

with $\Delta c_{i,j}(l)$ the l^{th} bit of $\Delta c_{i,j}$ that represents the difference between the l^{th} bits of c_i and c_j. Δf_k has thus been deduced from the difference of the selection function f_k (3), the latter providing the values of c_i and c_j.

4.4 Mean and Variance Analyses

Following (5), let us define $\{C_k|\Delta c_{i,j}(l) = 0\}$ and $\{C_k|\Delta c_{i,j}(l) = 1\}$ as the two subsets of coherence values for which the l^{th} bits of c_i and c_j are respectively equal and different. Because $\{C_k|\Delta c_{i,j}(l) = 0\}$ ($\{C_k|\Delta c_{i,j}(l) = 1\}$) gathers coherence values associated to pair of traces with a given bit having the same and

different HD or HW values, the expectation $E(C_k|\Delta c_{i,j}(l) = 0)$ should be higher than $E(C_k|\Delta c_{i,j}(l) = 1)$. By averaging these results for every bits from 1 to w, an adversary may expect disclosing the secret key using the following distinguisher :

$$\max_{k \in K} \left\{ \sum_{l=1}^{w} \left(E(C_k|\Delta c_{i,j}(l) = 0) - E(C_k|\Delta c_{i,j}(l) = 1) \right) \right\} \tag{6}$$

Similarly, the variances $V(C_k|\Delta c_{i,j}(l) = 0)$ and $V(C_k|\Delta c_{i,j}(l) = 1)$ should have greater values for wrong guesses than for the secret key, k_g. Thus, an adversary may also identify k_g with:

$$\min_{k \in K} \left\{ \sum_{l=1}^{w} V(C_k|\Delta c_{i,j}(l) = 0) \right\} \tag{7}$$

Let us denote by *mean+MSC* and *var+MSC* these two attacks afterwards.

4.5 Correlation Analysis

According to the adopted leakage model, the coherence of two traces is a decreasing function of the difference of $\Delta c_{i,j}(l)$. Assuming additionally that this function is linear, is equivalent to assume that the expectations of $(C_k| \sum_{l=1}^{w} \Delta c_{i,j}(l) = q)$ are decreasing with the increase value of q. This is to say that the more the words c_i and c_j associated to the two traces are different, the less these traces are similar. Thus, an adversary may analyse the correlation between C_k and $\sum_{l=1}^{w} \Delta c_{i,j}$ and identify the secret key by searching the guess with the maximum absolute score.

Let us denote by *corr+MSC* this attack afterwards.

4.6 Non-Parametric Tests

In 4.4, we explained why the probability density functions associated to the values $\{C_k|\Delta c_{i,j}(l) = 0\}$ or $\{C_k|\Delta c_{i,j}(l) = 1\}$ must have different values of expectation and variance for a correct guess of the secret key. Let us generalize this reasoning and more precisely let us assume:

- that for k_g, the secret key, the Cumulative Density Function (CDF) constructed with all values $\left\{C_{k_g}|\Delta c_{i,j}(l) = 0\right\}$ is unique and different from all the others,
- that for wrong guesses, k, the CDFs constructed with all values $\{C_k|\Delta c_{i,j}(l) = 0\}$ are similar in shape.

With these assumptions, the secret key can then be identified, using the Kolmogorov - Smirnov test [14, 15] of goodness of fit, with:

$$\max_{k_1 \in K} \left\{ \sum_{l=1}^{w} \sum_{k_2 \neq k_1} \delta \left(CDF[C_{k_1} | \Delta c_{i,j}(l) = 0], CDF[C_{k_2} | \Delta c_{i,j}(l) = 0] \right) \right\} \quad (8)$$

with:

$$\delta(F_1, F_2) = \frac{g_1 . g_2}{g_1 + g_2} . Sup_x \left| F_1(x) - F_2(x) \right| . \quad (9)$$

the maximal distance between F_1 and F_2, two Cumulative Distribution Functions.

Let us denote by *KS+MSC* this attack afterwards.

5 Experimental Results

In order to verify the efficiency of the SCAN, and of the leakage transformation with its derived attacks, we have applied them on a set of 5000 traces collected at the surface of an unprotected implementation of the DES. This FPGA implementation operates at 50 MHz. We also compared the results obtained with MSC based attacks to those of well known distinguishers. (It is to notice that the following attacks are noted with a P for power, to keep their usual name, however they are applied on electromagnetic curves.) Among them, we selected:

- The Bravais-Pearson correlation used in the time domain [3] and which is perfectly adapted to a linear leakage,
- the Difference of Means and more precisely the multi-bit Differential Power Analysis (DPA) that also works in the time domain [1] and sum the Difference of Means for each bit,
- the multi-bit DPA (DPAabs) which sums the absolute value of the Difference of Means of each bit [11],
- the Correlation Power Frequency Analysis (CPFA) described in [2] and further analysed [8], to provide a comparison with previous attempts to exploit the frequency domain,
- and two different implementations of the Mutual Information Analysis [6] based on kernel estimations. The first one (MIA) calculates the mutual information between the traces and the sum of all the output bits $\sum_{l=1}^{w} c_i(l)$ (see eq.3). The second one (MIA mb) calculates the mutual information between the traces and the values of each output bits $c_i(l)$ and then computes the average of all results.

Our evaluations have followed the framework proposed in [12]: we computed a global Success Rate taken on the eight sub-keys of the last round of the DES. It is to note that all EM traces were acquired with a Lecroy oscilloscope featuring a 20 GS/s sampling rate and using a low noise 63db amplifier with a 1 GHz bandwidth.

5.1 Efficiency : Number of Traces Required

Table 1 gives for each attack based on a HD model, the number of curves required to reach a given value of Success Rate. In this table 'mb' and 'word' allow identifying the attacks that work at the word level and the ones that work on each bit separately before combining the results obtained for all bits. As can be seen, all attacks are able to disclose the secret key with this limited set of traces. However, in this case, frequency domain analyses have given better results than time domain analyses, especially those applied after the transformation of the leakage.

It is to notice that the two MIA require less curves than CPA to reach a Success Rate of 80% and 100%. This may suggest that the leakage has a linear behavior but not only. It is also important to notice that all analyses based on the MSC, including the SCAN, are the only ones to reach a Success Rate of 80% after the processing of less than 1000 curves, while all time domain analyses have reach a Success Rate of $\sim$ 10%, only (except DPA abs). Additionally, one can note that all proposed analyses with the leakage transformation have allowed disclosing the key with fewer traces than the SCAN (50% less in the best case).

We can therefore conclude from Table 1 that analyses in the frequency domain may provide better results than time domain analyses and that transforming the traces does not suppress information but seems to increase significantly the number of informers.

Table 1. Number of processed traces vs Success Rate (HD model)

		Success Rate	10%	20%	40%	60%	80%	100%
time domain	word	CPA	775	1075	1525	2150	4475	5000
		MIA	1650	1850	2450	2900	3300	4150
	mb	DPA	850	1175	1750	2800	4250	4975
		DPAabs	500	550	720	825	1075	1400
		MIA mb	950	1150	1250	1600	1750	2100
frequency domain	word	CPFA	1110	1205	1410	1630	2025	3150
		corr+MSC	320	410	480	532	660	730
	mb	SCAN	375	390	420	480	615	1200
		mean+MSC	230	260	310	440	495	650
		var+MSC	440	450	535	670	780	1135
		KS+MSC	350	370	440	455	540	690

Table 2 gives the same results than Table 1 but this time in case of an adversary adopting the HW model. Only the MSC based analyses the DPA abs and one of the two MIA (MIA mb) are able to retrieve entirely the key with this set of 5000 traces. From Tables 1 and 2 we may thus conclude that MSC based analyses provide the best results. However that doesn't explain why these analyses outperform the time domain attacks. One first explanation could be that they work in the frequency domain, but this is not sufficient! Indeed, one of the MIA remains efficient and the DPA abs also.

Table 2. Number of processed traces vs Success Rate (HW model)

		Success Rate	10%	20%	40%	60%	80%	100%
time domain	word	CPA	fail	fail	fail	fail	fail	fail
		MIA	fail	fail	fail	fail	fail	fail
	mb	DPA	fail	fail	fail	fail	fail	fail
		DPAabs	2800	3175	3925	4400	4800	4950
		MIA mb	3550	3700	3900	4350	4550	4900
frequency domain	word	CPFA	fail	fail	fail	fail	fail	fail
		corr+MSC	2375	2515	2705	3495	3990	4810
	mb	SCAN	1750	1900	2300	2950	3625	4200
		mean+MSC	2430	2510	2685	3460	3980	4855
		var+MSC	4250	4400	fail	fail	fail	fail
		KS+MSC	2120	2580	3310	3710	4070	4495

5.2 Efficiency : CPU Times

It is necessary to notice that all our attacks were coded in C and were launched on a standard computer with a CPU running at 3 GHz. The CPU time costs of the different attacks were measured. Table 3 gives the results obtained for two sets of 500 and 1000 traces, respectively. The step refers to the number of traces between which the computation of the distinguisher is done to retrieve the key.

Table 3. CPU times of the attacks with a step of 10 curves

Number of traces :	500	1000
CPA	13s	26s
DPA and DPAabs	15s	30s
MIA	4m	8m
MIA mb	13m	25m
CPFA	13s	27s
SCAN	15s	31s
MSC based analyses	1h5m	4h20m

As can be seen, the application of a well implemented SCAN requires roughly the same CPU time than a CPA. However all MSC based analyses that work on pairs of curves are time consuming. Indeed, the increase of the number of informers comes at the cost of an increase of time computation which seems quadratic (for a set of n traces, the number of coherences to compute is proportional to $n.\frac{n-1}{2}$). Consequently, such attacks are to be used on a limited set of traces; i.e. on systems embedding a frequent refreshing of the keys.

5.3 Advantages of the Frequency Domain

One main advantage of working in the frequency domain is the ability to catch the leakage spread over time, while time domain attacks that aim at analysing

each time sample independently don't fully use it. Additionally, analysing several samples at a time, as frequency domain analyses do, may provide a certain level of robustness against the eventual existence of a time sample with an outlier behavior with respect to the leakage model. We verified this potential advantage. Figure 1 shows the results, obtained for the fourth sbox, of the multi-bit DPA based on a HD model after the processing of 1000 curves. As can be seen, there is a peak corresponding to a wrong guess of the key that prevents from finding the good key with 1000 curves while the SCAN, and the MSC based analyses, performed with 512 consecutive samples, are not disturbed by its occurrence as shown by Figure 2. This figure gives the coherence value obtained by each key hypothesis after the processing of the same 1000 traces with the SCAN. Thus, the 'filtering' of few peaks with an outlier behavior constitutes a first advantage of MSC based analyses.

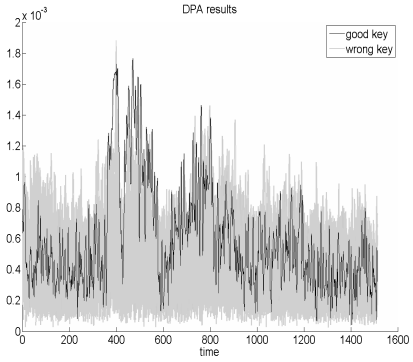

Fig. 1. Results of a DPA targeting the four output bits of the sbox $n^\circ 4$ after the processing of 1000 curves.

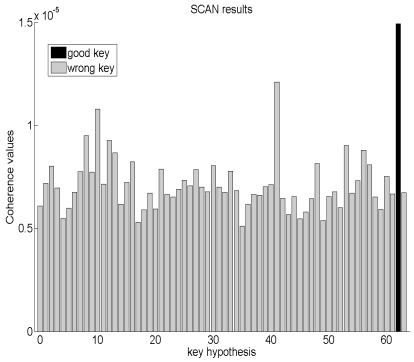

Fig. 2. Results of a SCAN targeting the four output bits of the sbox $n^\circ 4$ after the processing of 1000 curves.

The reading of Tables 1 and 2 highlights that the MIA and the analyses based on the Magnitude Squared Coherence give better results, on this set of traces, than the CPA when the HW and the HD models are adopted. Thus one may wonder about the correctness of the assumption according to which the leakage (an EM leakage in our case) depends linearly either on the Hamming Distance or the Hamming Weight, even if we were expecting to observe a strong linear dependency of the leakage with the HD because of the iterative implementation of the DES.

We thus tried to find the degree of the polynomial representing at best the evolution of the leakage according to the Hamming Distance and to the Hamming Weight. Figures 3.a and 3.b are scatter plots vs the HD and the HW of a single sample of the traces sorted according to the output value of the fourth sbox: $\sum_{l=1}^{w} c_i$. Figures 3.c and 3.d show the polynomial representing at best

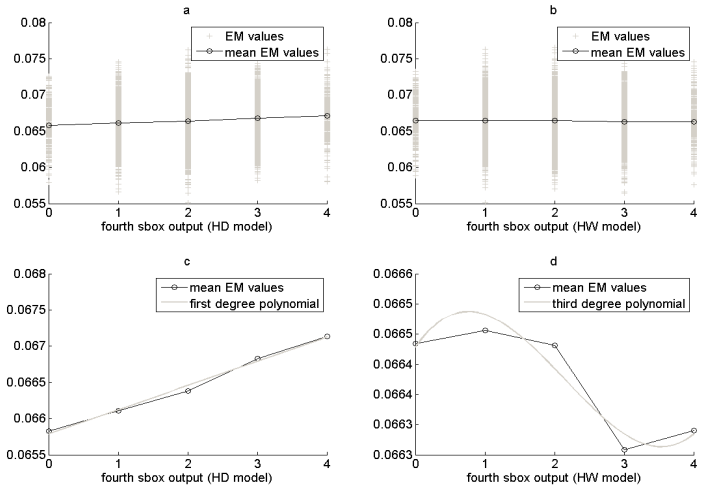

Fig. 3. Figures a and b show the values of a single sample of EM traces sorted according to guesses on the output of the fourth sbox. Figures c an d show the polynomials with the lowest degree representing at best the leakage.

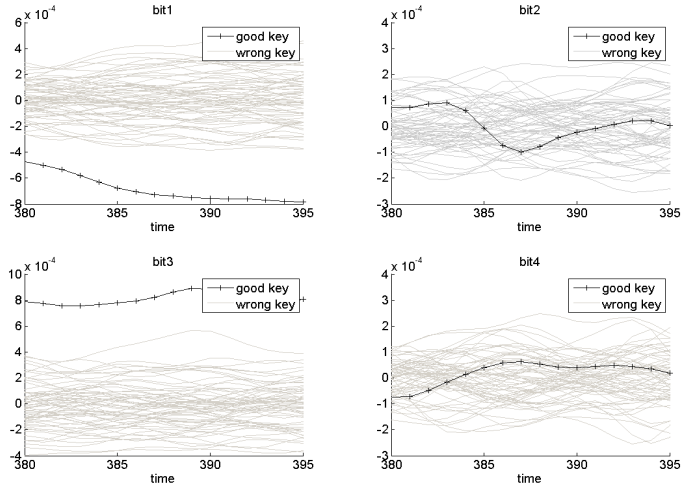

Fig. 4. Results of mono bit DPA targeting each of the four output bits of the fourth sbox.

the leakage. These polynomials were obtained with the least squared method. It should be noticed that similar figures were obtained for neighbouring samples.

The best modelling of leakage found when the traces are sorted according to the HD model, considered at the word level, is obviously linear as it was expected. This explains why analyses based on the HD models at word level, such as the CPA and the MIA, work well.

However, when traces are sorted according to a HW model, the leakage is clearly not linear, and a polynomial of degree three is at least needed to describe it correctly. It is to notice that the ordinates of Figures 3.c and 3.d are not the same, and that compared to 3.c, 3.d seems quite constant, suggesting that there is no leakage when working with $\sum_{l=1}^{w} c_i$. These observations explain why analyses based on the HW model at word level, such as the CPA and the MIA, do not succeed in disclosing the key.

On the contrary, when we observed each output bit separately (represented figure 4), results showed that the four output bits of the fourth sbox have different behaviors. Two of them don't seem to leak data dependent information, while the two others leak opposite information. These two last bits, when working at the sum level, cancel each other and this explains why the attacks, such as the multi-bits DPA, can't retrieve the key. However, attacks that work at the bit level, and give a positive value to the score of each bit, such as the DPA absolute Sum, the SCAN the MIA mb, are able to find the key.

From all the above analyses, we therefore conclude that attacks based on the MSC offer the advantages:

- to work at bit level and thus to offer a significant resistance to the eventual non linearity of the leakage model at word level,
- to score the leakage with a positive real value ranging between 0 and 1 so that the cancellation of the various bit contributions is avoided (DPA Absolute Sum and MIA mb do exactly the same),
- to be able to filter some peaks with outlier behaviors by working on several consecutive samples,

6 Conclusion

From all the above analyses and results, one may conclude that Magnitude Squared Coherence is an efficient tool for SCA. Indeed, it can be directly used as a distinguisher characterized by an interesting robustness to the occurrence of outlier behaviors on few samples of the leakage traces. Additionally, working with this distinguisher at bit level, confers a significant robustness against an eventual non linearity of the leakage at word level. The Magnitude Squared Coherence can also be used to transform a reduced set of traces into a wider set of scalar data without loss of information and even with a significant increase of the amount of information. The resulting set of data can then be advantageously used to obtain the secret key by statistical means. One may now wonder how to mount an higher order analysis with such a tool.

References

1. Régis Bevan and Erik Knudsen. Ways to enhance differential power analysis. In *ICISC*, pages 327–342, 2002.
2. E. Bohl, J. Hayek, O. Schimmel, P. Duplys, and W. Rosenstiel. Correlation power analysis in frequency domain. In *COSADE, Darmstadt, Germany*, 2010.
3. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *CHES*, pages 16–29, 2004.
4. Amine Dehbaoui, Sebastien Tiran, Philippe Maurine, Franois-Xavier Standaert, and Nicolas Veyrat-Charvillon. Spectral coherence analysis - first experimental results -. Cryptology ePrint Archive, Report 2011/056, 2011. `http://eprint.iacr.org/`.
5. Catherine H. Gebotys, Simon Ho, and C. C. Tiu. Em analysis of rijndael and ecc on a wireless java-based pda. In *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, pages 250–264, 2005.
6. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *CHES*, pages 426–442, 2008.
7. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.
8. Edgar Mateos and Catherine H. Gebotys. A new correlation frequency analysis of the side channel. In *WESS*, page 4, 2010.
9. Olivier Meynard, Denis Réal, Sylvain Guilley, Florent Flament, Jean-Luc Danger, and Frédéric Valette. Characterization of the electromagnetic side channel in frequency domain. In *Inscrypt*, pages 471–486, 2010.
10. P.D.Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short. In *IEEE Trans. Audio Electroacoustics*, pages 15:70–73, 1967.
11. François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede. Partition vs. comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected cmos devices. In *ICISC*, pages 253–267, 2008.
12. François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, pages 443–461, 2009.
13. Sebastien Tiran, Amine Dehbaoui, and Philippe Maurine. Magnitude squared coherence based sca. Cryptology ePrint Archive, Report 2012/077, 2012. `http://eprint.iacr.org/`.
14. Nicolas Veyrat-Charvillon and François-Xavier Standaert. Mutual information analysis: How, when and why? In *CHES*, pages 429–443, 2009.
15. Carolyn Whitnall, Elisabeth Oswald, and Luke Mather. An exploration of the kolmogorov-smirnov test as a competitor to mutual information analysis. In *CARDIS*, pages 234–251, 2011.

A SCAN Implementation

A.1 Naive Implementation of SCAN

During the application of a side-channel attack on a set of curves, one often wants to see the results progressively and not only after the processing of all the

available traces. This usually allows stopping an attack as soon as this one may be considered as successful. For that, an adversary have to compute the attack by steps, i.e. each time a given number of additional traces has been processed.

Algorithm 1 gives a first way of implementing the SCAN, based on a multi-bits DPA approach. It consists in computing a mean trace for each key hypothesis and for each bit of the predicted output. At each step, instead of doing the difference of means, the MSC is computed between the averaged traces associated to the two possible values of each predicted output bit.

However the computation of the Magnitude Squared Coherence implies the calculation of a significant number of Fast Fourier Transforms (see equations (1-2)). This number depends on the segmentation of the signals when the Welch method is adopted. For an algorithm such as the AES, retrieving the entire key implies the computation of the MSC for each 16 sbox, for each 256 possible key hypotheses and for each 8 output bits. Thus one have to compute 32768 MSC at each step.

A.2 Fast Implementation of SCAN

To reduce this number, in case of a low step value, one may compute directly the FFT of each trace. Due to the linearity of the Fourier Transform, he can then compute the mean of the Fourier Transform of the traces instead of calculating the mean trace. This result in a significant reduction of the CPU time needed to process a set of traces with a low step value. Algorithm 2 represents the pseudocode of this new approach.

It is to notice that the spectral components of the Fourier Transform of each traces are averaged. Thus the Magnitude Squared Coherence is computed on this mean instead of being computed at each frequency before being averaged. However experimental results have shown that these two methods lead approximately to the same results and enable to retrieve the key with nearly the same number of curves. Averaging the spectral components at the beginning greatly speeds up the algorithm as this reduces the number of MSC that have to be computed (one coherence on a mean frequency instead of a coherence at each frequency).

Algorithm 1 SCAN pseudocode

Input : messages T_i
Output : guessed key k_g*
for $i = 1$ **to** $nbcurves$ **do** // loop on the number of curves
  for $k = 0$ **to** nbk **do** // loop on the number of key hypothesis
    for $l = 1$ **to** w **do** // loop on the number of predicted output bits
      if $c_i(l) = 1$ **then** // value of a predicted output bit
        $T_k^{c_i(l)=1} += T_i$;
        $cpt_k^{c_i(l)=1}++$;
      else
        $T_k^{c_i(l)=0} += T_i$;
        $cpt_k^{c_i(l)=0}++$;
      end if
    end for
  end for
  if i mod step == 0 **then**
    for $k = 0$ **to** nbk **do**
      for $l = 1$ **to** w **do**
        $M_k^{c_i(l)=1} = T_k^{c_i(l)=1}/cpt_k^{c_i(l)=1}$;
        $M_k^{c_i(l)=0} = T_k^{c_i(l)=0}/cpt_k^{c_i(l)=0}$;
      end for
    end for
    $$k_g* = \min_{k \in K} \left\{ \frac{1}{w} \frac{1}{nbf} \sum_{l=1}^{w} \sum_{f} Coher((M_k^{c_i(l)=1}), (M_k^{c_i(l)=0}))(f) \right\}$$
  end if
end for

Algorithm 2 Fast SCAN pseudocode

Input : messages T_i
Output : guessed key k_g*
for $i = 1$ **to** $nbcurves$ **do**
  for $wind = 1$ **to** $nbwind$ **do** // loop on the number of sub-segments of the trace
    $F(wind) = \frac{1}{nbf} \sum_f FFT(T_{i,wind})(f)$;
  end for
  for $k = 0$ **to** nbk **do**
    for $l = 1$ **to** w **do**
      if $c_i(l) = 1$ **then**
        for $wind = 1$ **to** $nbwind$ **do**
          $F_k^{c_i(l)=1}(wind) += F(wind)$;
        end for
        $cpt_k^{c_i(l)=1} ++$;
      else
        for $wind = 1$ **to** $nbwind$ **do**
          $F_k^{c_i(l)=0}(wind) += F(wind)$;
        end for
        $cpt_k^{c_i(l)=0} ++$;
      end if
    end for
  end for
  if i mod step == 0 **then**
    for $k = 0$ **to** nbk **do**
      for $l = 1$ **to** w **do**
        $P_{1,0}^{k,c_i(l)} = \sum_{wind} F_k^{c_i(l)=1}(wind).F_k^{c_i(l)=0}(wind)$;
        $P_{1,1}^{k,c_i(l)} = \sum_{wind} F_k^{c_i(l)=1}(wind).F_k^{c_i(l)=1}(wind)$;
        $P_{0,0}^{k,c_i(l)} = \sum_{wind} F_k^{c_i(l)=0}(wind).F_k^{c_i(l)=0}(wind)$;
      end for
    end for
    $k_g^* = \min_{k \in K} \left\{ \frac{1}{w} \sum_{l=1}^{w} \left\{ \left| P_{1,0}^{k,c_i(l)} \right|^2 / (P_{1,1}^{k,c_i(l)}.P_{0,0}^{k,c_i(l)} \right\} \right\}$
  end if
end for