Improving First Order Differential Power Attacks Through Digital Signal Processing

Alessandro Barenghi
DEI–Dipartimento di
Elettronica e Informazione
Politecnico di Milano
Via Ponzio 34/5, 20133
Milano, Italy
barenghi@elet.polimi.it

Gerardo Pelosi
DIIMM–Dipartimento di
Ingegneria dell'Informazione e
Metodi Matematici
Università di Bergamo
Viale Marconi, 5, 24044
Dalmine (BG), Italy
gerardo.pelosi@unibg.it

Yannick Teglia
STMicroelectronics
ZI de Rousset, BP2, 13106
Rousset Cedex, France
Yannick.Teglia@st.com

ABSTRACT

Side-channel attacks pose a critical threat to the deployment of secure embedded systems. Both the academic and industrial communities have devoted considerable effort in order to evaluate the feasibility and effectiveness of these attacks. Differential-power analysis is a well-documented, powerful side-channel attack technique, which relies on measuring the power consumption of secure device (e.g. smart card, tamper-resistant circuits) while it computes a cryptographic primitive. These measurements are performed employing a digital oscilloscope and off the shelf components in order to tap into the power supply line of the device. Through correlating the power consumption of the device with the computed values, the technique aims at extracting the secret information from it, by means of exploiting the knowledge of the operations involving the cryptographic key which are performed in the algorithm. This topic has been widely discussed in the open literature, although there are no studies describing how to properly employ digital signal processing techniques in order to improve the effectiveness of the attacks. In fact, it is common use to employ the raw signal, exactly as it is recorded from the oscilloscope, only taking care of realigning properly the data sets on the time axis, while no attention is devoted to identify which part of the signal is actually carrying the correlated information. This paper presents a new pre-processing technique based on signal processing, effectively reducing the number of traces needed to perform a successful differential power attack by an order of magnitude with respect to the results obtained with raw datasets. This technique is validated through attacking a commercial 32-bit software implementation of AES running on a Cortex-M3 CPU, a widely deployed core targeted to embedded systems. All the previous open literature works on differential power analysis focused on 8 bit platforms due to the greater ease of leading the attack. The experimental results show that the proposed framework is effective in suppressing the noise in the power consumption recordings, without discarding any informative content. The main contribution of this paper lies in delineating a clear leakage model for software implemented cryptographic primitives and proposing an effective framework to extract it, thus showing the relevance of employing a proper pre-processing technique. The experimental results are evaluated through considering the level of statistical significance of the secret information extracted from the power consumption measurements, instead of examining only the absolute correlation values obtained from the analysis.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application Based Systems**]: Microprocessor/microcomputer applications; C.5.3 [**Computer System Implementation**]: Microcomputers—*portable devices*; E.3 [**Data Encryption**]: Standards (AES)

General Terms

Security

Keywords

Side-Channel Attacks, Differential Power Analysis

1. INTRODUCTION

The widespread use of embedded electronics has brought to the attention of the research community the threats represented by attackers having physical access to the devices that are in charge of providing secure operations for the user. The design issues related to secure hardware turn out to be more challenging compared with the ones of the standard circuits, since the usual design process does not consider detailed information about the possible side-channel vulnerabilities which will affect the complete product, such as electromagnetic emissions or unbalanced power consumption. Upon implementation, a cryptosystem runs on a device performing computations that combine input data and some secret (stored on chip in a non extractable way) in order to produce an enciphered output. For the sake of clarity, we will refer to these computations as encryptions, the input data as a plaintext, the internal secret as a key, and the output as a ciphertext. In this scenario, the attacker's aim is to infer the secret key stored inside the device. In a private-computation model, the attacker uses only input and output datasets, and the secrecy of the cryptographic key is demanded to the implementative choices concerning the cryptographic module. In side-channel attack scenario, the attacker is able to obtain extra leaked information related to the device computation. This additional information is extracted from a number of observable parameters (timing [15], electromagnetic radiation [3, 25], power consumption [16, 18]) which may be correlated with some portion of the key and may be exploited to infer the whole secret value. Typically, the instantaneous power consumption can be measured through inserting a probe in the external power supply connection of the device, and thus it represents a readily exploitable side-channel information supplier. The literature classifies the attacks, according to the statistical treatment applied to the side-channel data into "simple" (SPA) and "differential" (DPA) attacks [16]. SPA attacks involve visually interpreting power consumption measurements as a function of time in order to detect data-(in)dependent properties between the computed value and the power consumption, in precise intervals of time. DPA attacks rely on comparing the distribution of the measured power consumptions over a number of encryptions against a theoretical model (which depends on the structure of the circuit) in order to exploit data-dependent properties of the side-channel leakage.

This paper presents a pre-processing technique able to enhance the effectiveness of differential power attacks, demonstrating how to carefully filter the power traces with a methodology not reported in previous works, and makes a progress in evaluating the minimum number of traces needed to perform a successful attack. We validate our methodology using a 32-bit Cortex–M3 processor (a widely deployed core targeted to embedded systems) and a software implementation of AES-128 as experimental test setting, both without any power analysis countermeasure, thus limiting ourselves to first-order DPA attacks which exploits the highly local correlation of the secret key values with the power measurement records. Investigation of high-order attacks (HO-DPA), which correlate also multiple values of a measurement record (at different time instants) in order to circumvent some countermeasures [12, 24], is deferred to future works. The experimental campaign presents the first attack to a 32-bit software implementation of the AES: an attack environment widely recognized as affected by a large amount of noise, which hinders the attacks, with respect to the common 8-bit platforms, which,on the other hand, have been shown to be successfully attackable.
The paper is organized as follows: Section 2 gives a brief overview of related literature and states the relations of this work with them. Section 3 provides a short summary of the fundamentals of DPA attacks, presents the proposed pre-processing techniques and the interval confidence metric as a tool to evaluate the effectiveness of our method. Section 4 provides a description of the platform under attack and presents the experimental results. Finally, concluding remarks are drawn in Section 5.

2. RELATED WORK

Since its first appearance in [16, 18], the Differential Power Analysis (DPA) has emerged as a powerful attack technique, which allows the secret key of a cryptographic algorithm to be recovered through analyzing the measurements (power traces) of the feeding current of the computing device, in a non-invasive manner [28]. One of the most commonly targeted algorithms in side-channel attacks is the Advanced Encryption Standard (AES), due to its widespread adoption in a number of industrial and ISO standards. The open literature provides a wide range of attacks carried to both ASIC hardware implementations [21] and 8-bit software implementations on microcontrollers [17] or smart-cards [6, 11, 32] of AES. Evaluation of energy consumption and performance of various block ciphers on 32-bit processors is carried out in [10], while the resistance to side-channel attacks of a 32-bit processor with custom instruction set extension for AES implemented on FPGA has been investigated in [31], and compared with several implementation options. A related line of works investigated the power leakage exposure of Field-Programmable Gate Arrays (FPGAs) which offer features as functionality updates, re-configuration of the design, and non-recurring engineering costs that are advantageous with respect to ASIC designs in several applications. The power consumption of a SRAM-based FPGA [22, 30] does not significantly differ from the one of CMOS integrated circuits, nevertheless the various components of a FPGA (configurable logic blocks, programmable I/O blocks and routing logic) exhibit different power consumption features because of the different load capacitances. In [22], the authors provide evidence that implementations of elliptic curve cryptosystems on a Virtex XCV800 FPGA, without specific countermeasures, are vulnerable to SPA attacks; while the same platform is susceptible also to DPA attacks, as reported in [30] in the case of a DES implementation. In [29] the authors report a significant reduction in the power leakage of an AES implementation on FPGA, in case an unrolling and pipelining strategy is used to implement the algorithm. In [3, 25], the authors demonstrate the viability of electromagnetic attacks (EMA) on an 8-bit processor running at 4 MHz in a smart-card, whilst in [8, 9] the authors target a more complex device (a PDA) supporting mobile code applications running AES and elliptic curve cryptography. This is obtained performing a traditional differential attack (employing a Difference of Means test) in the frequency domain instead of working in the time domain as usually DPA attacks do. The approach proposed in [8, 9] has no similarity with ours, since the only point of contact is the use of Fourier transform, but no analysis of the significance of harmonic components with respect to the actual useful information modulated on the top of it is provided or exploited. In [4] the authors provide evidence of an attack to a simulated 8-bit `xor` gate array which may be used in the AES AddRoundKey primitive, and to a simulated 32-bit `xor` gate array which may be used in DES combination between the output of the Feistel function and the other half of the round state.

3. ATTACK METHODOLOGY

Following the description in [28], a DPA attack is mounted in five steps: i) select a key-dependent value to be observed; ii) collect power traces from the device while it is encrypting a number n of known plaintexts d_j, $j = 1, \dots, n$; iii) calculate the hypothetical intermediate values depending on every possible value of the selected key-portion; iv) select a

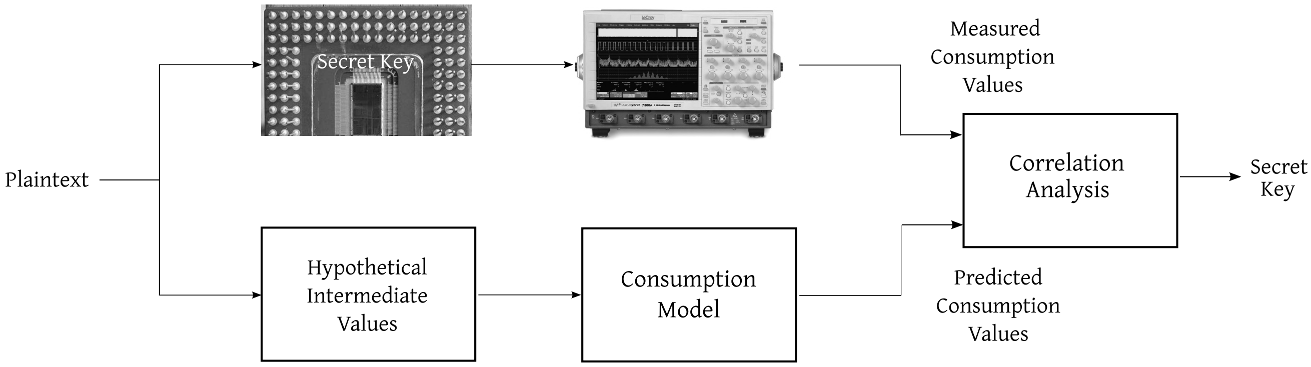


Figure 1: Differential power attack workflow

consumption model for the device and construct accordingly a *selection function* to classify the hypothesized consumptions; v) correlate the hypothetical consumptions predicted with the measured ones and select the hypothesis which fits best the measured behavior as the correct key candidate. The complete workflow of the DPA methodology is depicted in Figure 1. Indeed, the effectiveness of DPA attacks is attributable to the information leakage due to the *switching activity* triggered, at high level, by a known operation that has an outcome dependent from the specific value of a portion of the secret key. Since the device information leakage happens mostly in correspondence of the dynamic power dissipation caused by signal transitions in the underlying circuits, the consumption hypotheses are usually considering the switching activity of the parts of the circuit handling the key values: i.e., functional units and registers [23]. The usual choice in order to model the consumption in case a value is stored in a register, is to use the Hamming distance [4] between the previous and the new value stored by the latches. On the other hand, employing the Hamming distance as a metric implies that the attacker either knows or is able to guess which values employed in the algorithm are actually stored in latches and which are computed on the fly: this in turn implies the knowledge of some either hardware or software implementation details of the cryptosystem. Another possibility is to derive the model through the construction of a template consumption profile [27] through measuring the behavior of a device for all the key hypotheses, and subsequently correlate it to the functioning of the device under attack. This requires the attacker to be in possess of an exact copy of the device under attack where he can arbitrarily choose the key employed in the encryption. This approach enables the attacker to obtain the best model of the device, also taking into account the noise produced, although it is required to collect a significant amount of traces in order to properly model both the leakage and the noise produced by the device. In the case under exam, we carried out the attack with both the Hamming weight and the Hamming distance model, since we were in full knowledge of the software implementation details and we had a clear description of the hardware structure. Since the results employing the Hamming distance model were nearly as efficient as the ones obtained with the Hamming weight, we chose to present the ones which are obtainable by an attacker who has no knowledge of the detailed information in our possess (see Section 4).

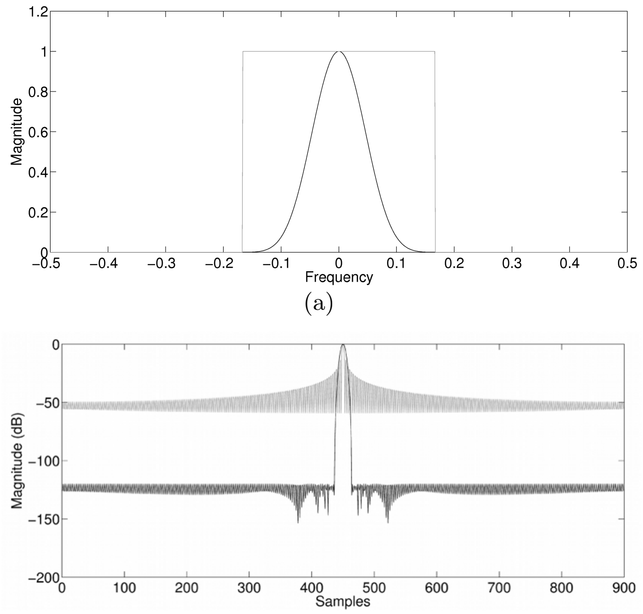
Our pre-processing methodology aims at selectively extracting the useful information from the input traces, which we will use to correlate each filtered trace with the hypothetical power consumption estimated for each subkey value employing the Pearson's coefficient as the correlation metric [4]. The proposed framework computes the frequency spectrum (DFT) of the power traces, designs a passband digital filter (identifying which parts of the spectrum are carrying the useful information), filters each trace multiplying its frequency spectrum by the spectrum of the designed filter, applies the inverse Fourier transform (IDFT) to get back a set of time-domain signals, and applies Pearson's analysis to the filtered power traces. The pre-processing discards the noise which is not correlated with the switching activity through proper signal processing, thus providing traces belonging to a simpler model which is both closer to the theoretical one, and easier to estimate during a possible template attack.

3.1 Signal Processing of Power Traces

The frequency spectrum can be generated via a discrete Fourier transform (DFT) of the trace samples, and the resulting values are usually presented as amplitude and phase, both plotted against frequency. The amplitude values represent the magnitude of each harmonic component, while the phase values represent its time shifting. The device information leakage happens mostly in correspondence of the dynamic power dissipation caused by signal transitions in the underlying circuits.

The most significant part of dynamic power dissipation in CMOS circuits [23, 28, 34] is proportional to capacitance, voltage swing and operating frequency of the considered resource. At physical level this dynamic power dissipation is driven by the working clock frequency, which provides a synchronization signal for the start of the switching activities of the logic gates. Thus, we expect that the harmonics composing the working clock signal show a significant magnitude with respect to other frequency components. The information leakage is amplitude modulated on these harmonics, which act as carriers for the actual value of the dynamical power consumption of the circuit. In the case the circuit is driven by different clock trees with different working frequencies (e.g., external buses are usually driven with a clock splitter), multiple spikes in correspondence of frequency bands centered around these working frequencies, are expected. Since in the time-domain the contribution of all the harmonics is added together, it is reasonable to assume that the eviction of signal components which are unrelated with the useful computation will yield a signal denser in informative content related to the key manipulation, due to the elimination of polluting noise.

The traditional signal processing approach to selectively extract only a subset of the harmonic components, is to design a multi-bandpass filter. Since the elaboration of sampled signals involves the use of the discrete Fourier transform, the time domain traces have to be considered as periodic (with period equal to the number of recorded samples) in order to correctly read the frequency response samples. A good bandpass filter should be able to completely discard the unwanted part of the input signal spectrum (blockband), while keeping the amplitudes of the selected parts (passband) as close as possible to the original ones. Thus, the ideal shape of a finite impulse response (FIR) filter for this purpose is a rectangle (rectangular window): however in this case the corresponding time representation of the filter response would result in an *aliased sinc function*[1] [20].
Thus, if a rectangle window is employed, the irregular ripples of the aliased sinc would introduce unwanted artifacts in the time domain, which would in turn pollute the trace with fabricated peaks. An optimal window set by the filter should mimic a rectangular shape in the passband as close as possible, even though maintaining a finite non-aliased impulse response. The selection criteria for the choice of a proper window depends on the actual needs of the specific problem. In the transformed domain, window functions like Hamming, Hann, Blackman-Harris, and rectangular [20] exhibit a fixed dependency between the reduction of side-lobe ripples and the consequent loss of resolution (main-lobe widening). This dependency does not allow us to reduce the entity of the side-lobes without losing accuracy in the filtered power traces and thus we did not consider these windows as a reasonable choice. The window shapes which allow the designer to tune at will the entity of the side-lobes roll-off, while keeping a high precision, are the Chebyshev[2] and the Kaiser[3] windows [20]. The Chebyshev window provides (in the transformed domain) constant side-lobe peak levels, whilst the Kaiser window's provides a rapid sidelobe roll-off at the cost of having the amplitude of the first lobes greater than the corresponding lobes on a Chebyshev's window with the same width. Deeming more important the uniform reduction of the sidelobes level, the chosen window shape is the Chebyshev one, which keeps all sidelobes in the transformed domain (time in our case) equally under a precise threshold attenuation (design parameter) thus reducing the aliasing effects at will. In our design choice the Chebyshev window was designed to have the sidelobes de-amplified by 120 dB, thus maintaining the artifacts added to the samples within the range of the numerical error of the computer computing the attack algorithm. By comparison, the choice of a rectangular window of the same amplitude, as depicted in Figure 2(b), would have caused aliasing effect as high as a third of the signal, since the first ripples are as high as -10 dB, altering the traces in a non negligible way. As depicted in Figure 2(a), the Chebyshev window has full amplitude only at the center of its passband and rapidly decays to zero. This does not represent a problem in our methodology since we are considering the leakage to be concentrated around very specific frequencies, thus we are not hindered by a narrow passband region.



(a)

(b)

Figure 2: Comparison of both the frequency spectra (a) and the time-domain impulse responses (b) of a rectangular and a Chebyshev window

The final multi-bandpass filter is built through placing on each selected frequency a Chebyshev window and through taking care of replicating the same window pattern along the samples of frequency domain to obtain a real and even discrete spectrum. A real and even spectrum (w.r.t. either the DC component or the half of the Nyquist frequency) has the property to have a zero/linear phase (linear phase if a causal filter is preferred). The zero-phase (linear-phase) property implies that the delay introduced by the filter on all the harmonics of the input signal is the same, thus the filter does not cause *delay distortion*. This, in turn, implies that there is no dilution of the informative content due the mutual de-synchronization of the harmonics, which would decrease both the effectiveness and efficacy of the attack. The use of the discrete Fourier transform implies that in the time domain the convolution between the input signal and the FIR is computed through a circular convolution, instead of the usual linear convolution. Thus, to obtain the same filtering effect as a linear convolution, it is mandatory to pad both the input signal with a batch of zeros as long as the width of the impulse response of the filter, and the impulse response with a batch of zero samples as long as the number of samples of the input signal.

[1] $rect_M(k) = 1,\ -\frac{M-1}{2} \leq k \leq \frac{M-1}{2} \overset{IDFT}{\leftrightarrow} \frac{sin(M\pi n/N)}{sin(\pi n/N)}$; $k, n \in \{-\frac{N}{2}, \ldots,\ \frac{N}{2}\}$

[2] $w(k)_{Chebyshev} = \frac{\cos\left(N\cos^{-1}\left(\alpha\cos(\pi\frac{m}{k})\right)\right)}{\cosh\left(N\cosh^{-1}(\alpha)\right)}$, where $\alpha = \cosh\left(\frac{1}{N}\cosh^{-1}(10^{\gamma})\right)$, $m = 0, \ldots, N-1$, $k \in \{-\frac{N}{2}, \ldots,\ \frac{N}{2}\}$

[3] $w(k)_{Kaiser} = \frac{I_0(\beta\sqrt{1-(\frac{k-\rho}{\rho})^2})}{I_0(\beta)}$, with $\rho = \frac{N-1}{2}$, I_0: 0^{th} order modified Bessel function of the first kind, $k \in \{-\frac{N}{2}, \ldots,\ \frac{N}{2}\}$

3.2 Correlation Analysis

After filtering the traces through the aforementioned procedure, we are willing to evaluate the effectiveness of the improvement in the attack. We decided to rely on Pearson's correlation coefficient analysis, known as CPA, to perform the attacks to a software implementation of AES-128 running on a 32-bit Cortex–M3 CPU.

The sample correlation coefficient r is a biased estimator of the true Pearson correlation coefficient ρ between two random variables when both of them are normally distributed. The expected value of correlation coefficient based on random sampling, of size n, from a normal population is approximately $\mathbb{E}[r] = \rho\left(1 - \frac{1-\rho^2}{2n}\right)$ with an even more exact result given by an infinite series containing terms of smaller magnitude. Elaborating the previous equation, the recommended unbiased estimator for the correlation coefficient is obtained as: $\hat{\rho} = r\left(1 + \frac{1-r^2}{2(n-3)}\right)$ [19]. In the settings of a DPA attack, the number of sample correlations is always relatively high (greater than 50) therefore, the bias is usually ignored assuming $\hat{\rho} = r$.

In order to assess the degree of statistical significance for a value of r, it is important to evaluate a confidence interval I_r for the hypotheses that the actual unknown Pearson's correlation ρ is equal to the computed value of the sample correlation. Consequently, the likelihood of the interval I_r containing the true ρ value, is chosen through selecting a confidence level $\gamma = 1-\alpha$ such that $\mathrm{Prob}\{\hat{\rho} \in [r_l,\ r_u]\} \geq \gamma$, thus determining the position of the interval boundaries $I_r = [r_l,\ r_u]$. This interval widens as the desired confidence level increases. The confidence interval I_r is therefore a metric of the statistical significance of the computed value r, which is the correlation index employed in the CPA procedure to distinguish the correct key hypothesis from the other ones.

For each time instant $\bar{t}$, given n sampled values of consumption for different plaintexts d_j and fixed a key-portion value $\bar{k}$, the $r_{\bar{t},\bar{k}}$ correlation coefficient for that key-portion is computed between the predicted consumptions for each plaintexts and the ones actually measured experimentally. A key hypothesis is deemed to be the best candidate when the estimated r surpasses the one of the other hypotheses for a time interval. A best candidate key hypothesis which holds also when raising the number of traces, n, at will, is deemed to be the correct key hypothesis. Since comparing two different $r_{\bar{t},\bar{k_1}}$, $r_{\bar{t},\bar{k_2}}$ values lacks statistical significance we will use as a metric of the effectiveness of the attacks: the lack of overlapping of the two confidence intervals $I_{r_{\bar{t},\bar{k_1}}}$, $I_{r_{\bar{t},\bar{k_2}}}$ for a defined confidence level γ. This lack of overlapping implies that the value of $r_{\bar{t},\bar{k_1}}$ is different than the one of $r_{\bar{t},\bar{k_2}}$ with at least probability γ.

Thus a quantitative and robust evaluation of the minimum number of traces n_{min} determining the success of an attack, is done through taking as n_{min} the least number of traces for which the confidence interval of the correct key is definitely above the one of the best wrong hypothesis. This evaluation will be used in Section 4.3 to state the improvements obtained through the pre-processing (filtering) of traces.

Since the variance of the sample correlation r is dependent on both sample size n and ρ size, it is not possible to calculate directly the confidence limits $[r_l,\ r_u]$ of the interval I_r. An indirect method to compute the interval size is to employ the so-called r to Z (also known as Fisher's) transformation: $Z = \frac{1}{2}\ln\frac{1+r}{1-r} = \mathrm{arctanh}(r)$. The obtained sample coefficient Z results to be normally distributed with mean $\left(\mathrm{arctanh}(r) + \frac{\rho}{2(n-1)}\right)$ and variance $\frac{1}{n-3}$. More specifically, the sample distribution of Z tends to be normal as quickly as the sample size increases, for any values of ρ. In order to compute a confidence interval for a given value of the sample correlation r and a fixed sample size n, at first, the confidence limits $[\xi_l,\ \xi_u]$ for the Z sample coefficient are computed: $\xi_l = z_r - \frac{z_{1-\alpha/2}}{\sqrt{n-3}}$, $\xi_u = z_r + \frac{z_{1+\alpha/2}}{\sqrt{n-3}}$, where $z_r = \mathrm{arctanh}(r)$, and $z_{1\pm\alpha/2}$ is the result of the quantile function of the standard normal distribution evaluated in $(1 \pm \alpha/2)$. Subsequently, the computed confidence limits of ξ_l and ξ_u are transformed back to derive the confidence interval $I_r =[r_l,\ r_u]$ as: $I_r =[\tanh(\xi_l),\ \tanh(\xi_u)]$.

4. FRAMEWORK EVALUATION

In this section we put into practice the attack framework described in Section 3. In order to validate our proposed filtering technique we chose to attack a software implementation of AES running on a 32-bit processor, targeting the output of the first SubBytes operation byte-wise. This implementation for the AES algorithm is expected to be affected by a high level of noise, due to the whole 32-bit registers and logic switching at the same time. To the best of our knowledge, this is the first successful attack since previous works focused either simulated 32-bit processors [4, 31] or on Difference of Means based attacks employing EMA [5, 8, 9].

4.1 Device Architecture

The chosen target platform is an ASIC implementation of a Cortex-M3 core [1] using a 180 nm process resulting in a maximum working frequency of 72 MHz. This CPU, belonging to the latest ARM architecture, the ARMv7-M [2], is a 32-bit processor, using Harvard architecture with a single 3 stage pipeline with branch speculation. The implementation chosen for the attack is equipped with 512 kB of on-die flash memory, used for code and constants storage, 64 kB of SRAM used for the data elaboration and a USB, USART and I^{2}C bus and a controller for SD card memory management, mounted on a standard development board, not an ad-hoc designed for DPA[4]. The implementation of the chip is not endowed with hardware power attack countermeasures, since these are not usually included in these chips and their evaluation is beyond the scope of this article. The clock source for the chip is an external quartz oscillator running at 8 MHz. All the required clock frequencies for the chip are obtained through a clock splitter, which generates both the core frequency and multiple split ones for the communication buses.

This processor is oriented towards the industrial market and is used mainly in wireless telecommunications and industrial control scenarios. This model has been widely deployed in the last years, and is available by a number of the main suppliers in the market, thus it represents a significant benchmark platform with no known DPA attacks. The attacked AES-128 is a straightforward `C` implementation of the AES algorithm employing a single precomputed S-Box to perform the SubBytes operation, without power analysis countermeasures. The source code is compiled with an ARM,

[4] Further details on the board model cannot be disclosed due to confidentiality issues

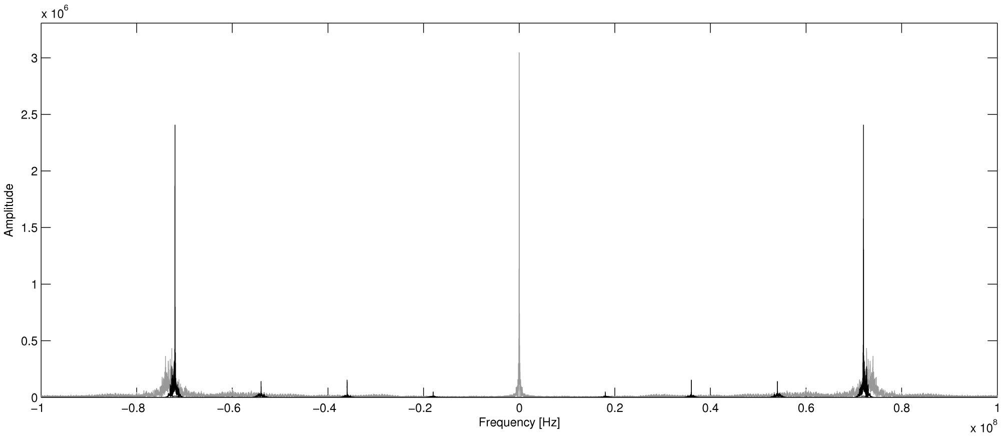


Figure 3: DFT spectrum of a measured trace (in gray) superimposed with the multi-passband filter (in black); the x-axis values range within $[-f_s/2,\ f_s/2]$ (where $f_s = 5$ GHz is the sampling frequency), for the sake of readability the picture does not shows the harmonics out of $[-\frac{1}{25}f_s/2,\ \frac{1}{25}f_s/2]$ since they do not provide any useful information)

cross compile toolchain based on GCC, and with the optimization level set to -O2 to achieve optimal performances.

4.2 Workbench Description

The employed oscilloscope is a LeCroy Waverunner Wave-Pro 7100A with a maximum sampling rate of 20 Gsamples/s and a 1 GHz analog bandwidth at -3 dB. The employed active differential probe is a LeCroy AP034 (1 GHz analog bandwidth at -3 dB, Common Mode Rejection Ratio above 80 dB) connected to a 2 Ω shunt inserted on the only power supply line available for the Cortex-M3 SoC. All the measurements have been taken with a 5 Gs/s sampling frequency in order to fully exploit the oscilloscope and the probe warranted bandpass regions and taking a reasonable safety margin. The digital to analog converter in the oscilloscope discretizes the signal over 256 level and saves each sample to a single signed byte integer value, thus achieving a typical 48.16 dB Signal-to-Quantization-Noise Ratio. The measurement for each trace was obtained averaging over 64 executions of the encryption of the same plaintext and stored directly on the oscilloscope mass storage for further retrieval. This was done in order to reduce the instrumental measurement noise, which, being well modelled by a random variable following a Gaussian distribution with zero mean, will average out in different realizations. On the other hand, averaging has no impact on the noise caused by either the intrinsic architectural structure of the chip or the implementative strategies of the attacked algorithm. From now on, we shall indicate with *trace* the result of averaging 64 sampled traces collected while encoding the same plaintext. The starting and ending points for the measurement were determined by a software risen trigger on one General Purpose Input/Output (GPIO) pin of the chip, asserting the pin before the algorithm started and deasserting it after the end of the execution. We acquired a total of 10000 traces, each one using 196 kB of disk space, resulting in a total disk occupation of 1.9 GB without the use of any compression techniques. The trace pre-processing phase has been done with custom scripts run in Matlab-2009a on an Intel, Core, 2 Quad Q6600 at 2.4 GHz endowed with 4 Gb of DDR2 DRAM and the time required for a full filtering campaign is approximately 10 minutes. The DPA attack has been performed employing the Matlab scripts available from the OpenSCA toolkit [7] and requires roughly one and a half minute to perform an attack attempt with all the 10000 traces.

4.3 Experimental Results

In order to design the filter for the attack, we first computed the Fourier transform of a trace, shown in grey in Figure 3. Subsequently, we checked that the main peak appearing is in correspondence with the core clock frequency of the CPU, as expected from the fact that the main consumption contribution comes from the computing core. After locating both the main clock and the significant harmonics generated by the clock splitter on the spectrum, we placed the passband windows centering them on the peaks and designing them to have a 500 kHz width to compensate for environmental fluctuations in the signal, without including the noise from the frequencies near the clock. The regions of the bandpass windows are depicted in black over the spectrum in Figure 3. Even though there may be other periodic behaviors in the execution of a cryptographic primitive (for instance, due to the loop based structure of block ciphers), the spectral components related to those loops may safely be discarded, since the source of the leakage (i.e., the gates switching activities) is driven by the clock, and thus is not influenced by such iterative structures. The unfiltered versus filtered signal power ratio ($G_{dB} = 10\log\frac{A_{unfiltered}}{A_{filtered}}$, where A denotes the sum of the modules of every value in the DFT squared) has an average value of $G_{dB} = 4.88$ dB over the traces. Figures 4(a) and 4(b) depict a sample trace respectively before and after the pre-processing through filtering. The spikes present in Figure 4(a) are due to the effect of the auxiliary input-output circuit employed to trigger the start of the acquisition on the oscilloscope. After applying the filter, both the effects of the trigger and the slow oscillation in the original signal are completely removed, together with the high frequency noise. The resulting trace is also free

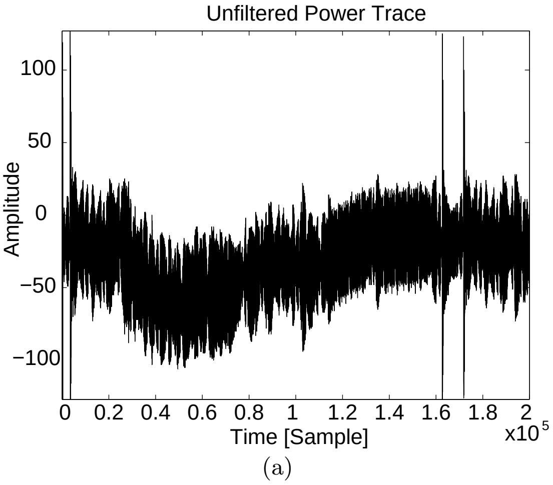


Figure 4: A sample trace pre (a), and post (b) filtering

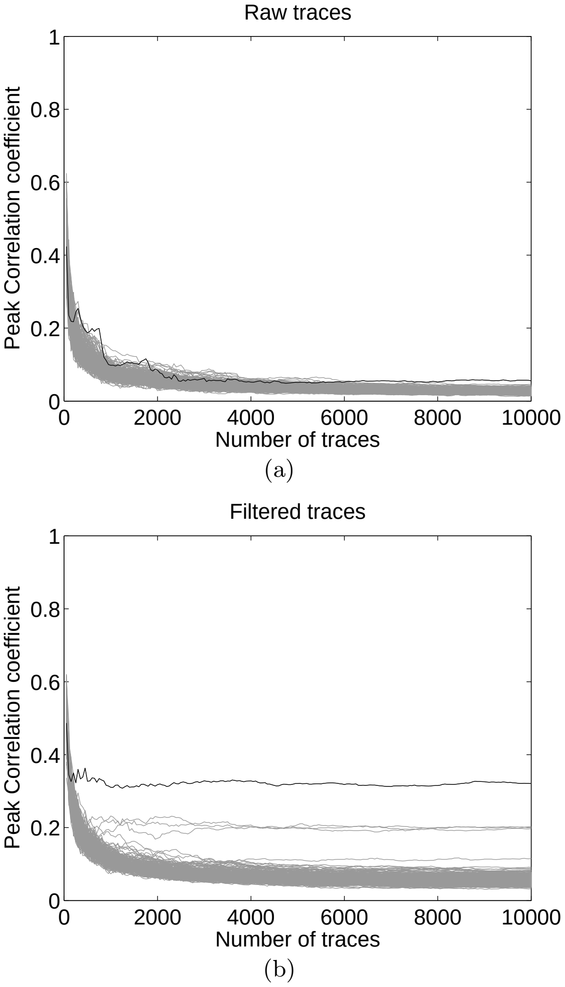


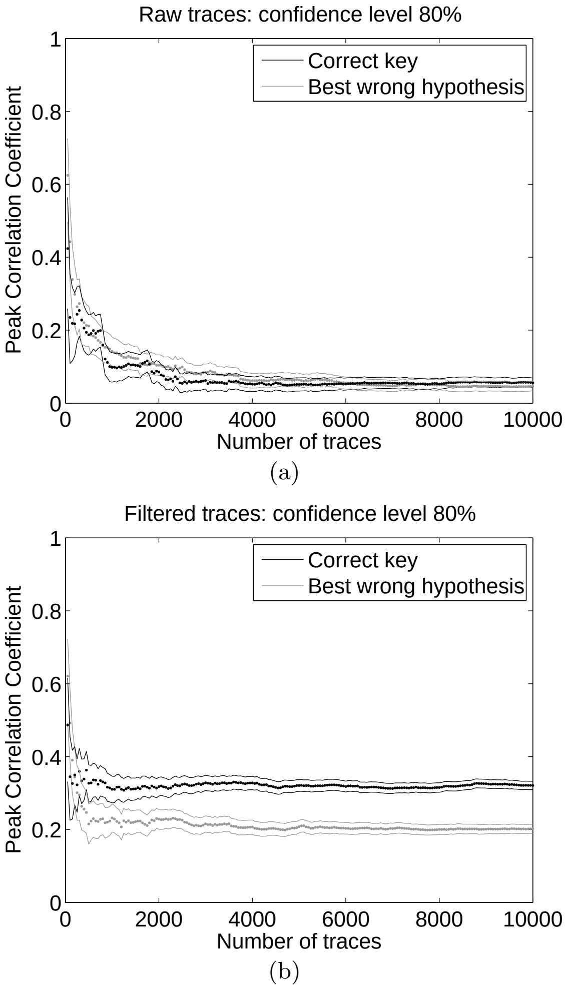
Figure 5: Correlation coefficients of the different single-byte key hypotheses, correct key shown in black: raw traces (a), filtered traces (b)

from any constant DC shift and thus does not need an additional average removal phase. The post processed trace (see Figure 4(b)) shows clearer features, and a regular behavior which enhances the possibility of locating the time interval in which the encryption is computed through direct optical inspection and provides ground for a simpler temporal alignment of the traces.

Figures 5(a) 5(b) depict the result of successfully running the attack on the filtered and unfiltered traces in terms of the maximum absolute value of the correlation coefficient for each key hypothesis (correct one in black) with an increasing number of traces, regardless of the confidence level of the result obtained. The pictures report the result for attack on the first byte of the key: the attacks on the remaining bytes show results within a negligible variation range (less than 1%). For the sake of clarity, from now on, all the results will be presented on a single byte attack.

It is evident from Figure 5(b) how, after the filtering, the correlation value related to the correct key hypothesis is set apart from the wrong ones starting from a low number of traces (roughly 300) and its value shows almost no fluctuations w.r.t. the growth of the employed traces. By contrast, in Figure 5(a) the estimated r value for the correct key hypothesis emerges using a higher number of traces (approximately 6000) and does not show the same stationary behavior as the one obtained through filtered traces.

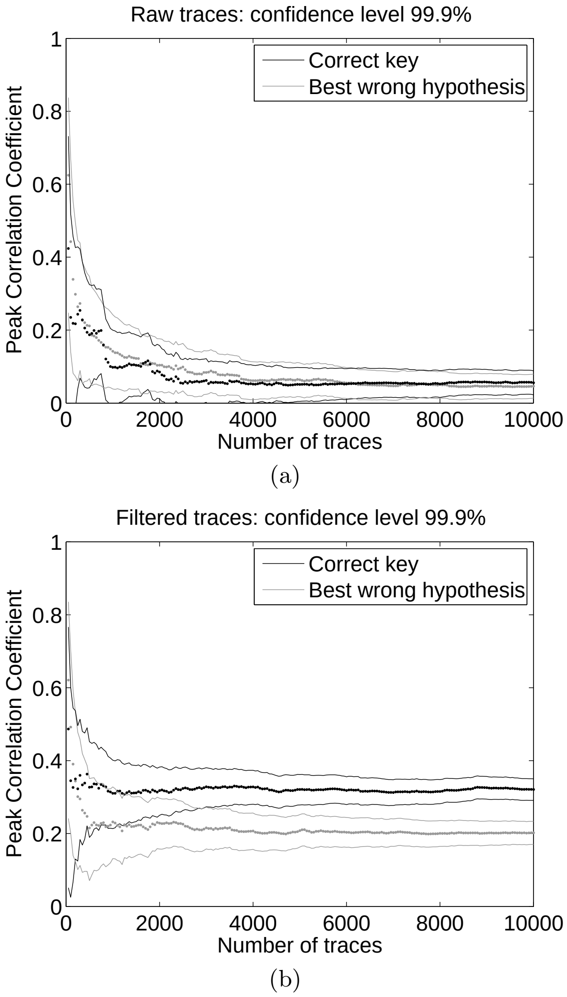
In order to further evaluate the soundness of the advantage of applying our pre-processing technique, we analyze the confidence intervals for the estimated values of the correlation coefficient r, comparing the one of the correct key (depicted in dotted black) against the one of the best wrong guess. In Figure 6 and Figure 7, the dotted grey line represents the wrong key hypothesis with the highest probability of being mistaken as the right one, for each attack attempt. Figures 6(a) 6(b) depict the width of the confidence intervals, with a confidence level of 80%, for the maximum estimated correlation coefficients, while Figures 7(a) 7(b) are depicting the same confidence intervals with γ =99.9%. The confidence intervals in Figure 6(a) show that the attack is feasible with unfiltered traces, albeit with a thin confidence margin. In fact, in the part where the correlation coefficient of the correct key dominates steadily the others (from



(a)

(b)

Figure 6: Correlation analysis of unfiltered (a) and filtered (b) traces, assuming a confidence level γ =80%



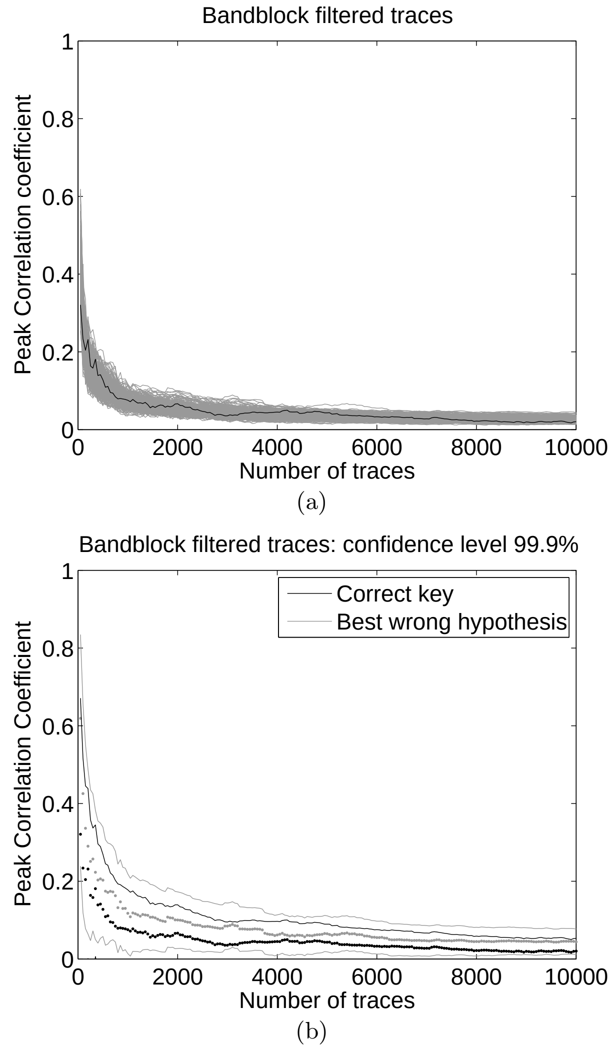
(a)

(b)

Figure 7: Correlation analysis of unfiltered (a) and filtered (b) traces, assuming a confidence level γ =99%

6000 traces onwards), the two confidence intervals are still overlapped, thus implying that the estimate does not reach an 80% confidence level. On the other hand, employing the filtered traces the statistical significance required for the estimate is met with only 450 traces, point at which the two confidence intervals stop overlapping, as shown in Figure 6(b). Willing to analyze the attack with a higher confidence level, the Figures 7(a) 7(b) show how, with a confidence level as high as 99.9%, the attack with the filtered traces yields positive results, with the two confidence intervals being clearly non-overlapped from 3000 traces onwards. On the other hand, the estimate obtained through using the unfiltered traces is not able to provide an estimate with this confidence level, since the two intervals are almost completely overlapping even when employing 10000 traces.

In order to verify that the discarded harmonics contained negligible informative content, we conducted the same attack employing traces obtained through keeping only the noise we discarded before and blocking all the informative harmonics. This is also commonly denominated *bandblock* filter. The results of the attack, depicted in Figures 8(a) and 8(b), show that the discarded harmonics do not carry any practically useful informative content to lead a first order DPA. In particular, the confidence intervals for the value of the correct key shows a fluctuating trend which tends to stabilize slightly lower than the best wrong guess, pointing to a failure in the attack. We also point out that the proposed pre-processing technique has an additional advantage: it allows the attacker to reduce sensibly the amount of disk space used. If only the non zero coefficients of the DFT are saved on disk and the traces are anti-transformed on the fly before the attack, the reduction in occupied disk space is directly proportional to the amount of discarded harmonics. This enables an attacker to tackle devices with a very low information leakage, since it is possible to store more traces with the current data storage capabilities. This pre-processing may be easily run directly on PC-based oscilloscopes during the capture phase through implementing the multi-bandpass filter directly in `C` since the procedure is particularly lightweight. In the illustrated case, this reduces the occupied diskspace to 0.4% of the original 1.9 GB, i.e. 64 MB.



(a)

(b)

Figure 8: Traces filtered using the bandblock filter: correlation coefficients of the different key hypotheses, correct key shown in black (a); confidence intervals for the correlation coefficient of the best wrong key hypothesis and the correct one (b)

5. CONCLUDING REMARKS

This paper has presented a new pre-processing technique based on multi-bandpass filters which enhances the informative density of the traces. This represents a pioneering study in the application of digital signal processing techniques to power traces in order to enhance the efficiency and effectiveness of first order DPA attacks. The technique has been validated experimentally through attacking a 32-bit software implementation of AES, lowering the minimum number of traces and reducing the number of traces needed to lead an attack with a reasonable confidence margin from 6000 to 450, thus reducing the quantity of computation and disk storage needed. This is the first result obtained against a full 32-bit CPU, which has a very wide deploy-base in the current market. The analysis of the results on the filtered traces, has shown that the informative content leaked is carried mostly by the clock frequencies driving the different parts of the chip, meeting the expectations inferrable from the theoretical consumption model of digital logic circuits. The proper filtering of the traces allows also an easier application of realignment techniques, since it removes artifacts not related to the information needed to carry out the attack. This methodology can also enhance the efficiency of template attacks, since the model of the circuit, which must be inferred from the traces, is simplified after the filtering, without any loss in the efficiency. The spectral analysis of the traces also determined that it is not necessary to acquire them with a sampling frequency exceeding twice the one of the fastest portion of the device handling the cryptographic computation, thus implying that less precise equipment than the one in our possession can be successfully used to mount the attack. On the other hand, employing apriori less precise equipment entails the possibility of missing the leakage from unknown parts of the architecture running at higher clock rates or from architectural glitches that have a faster transition period. Further developments of the technique include the possibility to apply a systematic approach and automatically infer the clock frequency of the parts of the circuit that are leaking, through performing multiple attacks employing traces filtered only on a single portion of the spectrum. Moreover, it is possible to further refine the part of the spectrum through parametrizing the filter passbands widths and positions (e.g. through using a binary decision tree algorithm to spot and select the most informative parts of the spectrum), using as a figure of merit for the refinement, the ratio between the obtained coefficients of the best key candidate and the second best, or any other information metric.

6. ACKNOWLEDGMENTS

The authors wish to thank Guido Bertoni, Luca Breveglieri, and Bernard Kasser for their valuable comments on preliminary versions of this paper. The work of Gerardo Pelosi was partially supported by the Italian MIUR, within PRIN SESAME 2007 project.

7. REFERENCES

[1] Acorn Risc Machines (ARM). Cortex^{TM}M3 Processor, February 2010. http://www.arm.com/products/processors/cortex-m/cortex-m3.php.

[2] Acorn Risc Machines (ARM). Cortex^{TM}M3 Technical Reference Manual, February 2010. http://infocenter.arm.com/help/.

[3] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side-Channel(s). In Kaliski Jr. et al. [14], pages 29–45.

[4] E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In Joye and Quisquater [13], pages 16–29.

[5] C. H. Gebotys and B. A. White. EM Analysis of a Wireless Java-based PDA. *ACM Trans. Embed. Comput. Syst.*, 7(4):1–28, 2008.

[6] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. Towards Sound Approaches to Counteract Power Analysis Attacks. In Wiener [33], pages 398–412.

[7] E. Oswald. OpenSCA, An Open Source Toolbox for Matlab, February 2010. http://www.cs.bris.ac.uk/home/eoswald/opensca.html.

[8] C. H. Gebotys, S. Ho, and C. C. Tiu. EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA. In Rao and Sunar [26], pages 250–264.

[9] C. H. Gebotys and B. A. White. Methodology for attack on a Java-based PDA. In *CODES+ISSS '06: Proceedings of the 4th international conference on Hardware/software codesign and system synthesis*, pages 94–99, New York, NY, USA, 2006. ACM.

[10] J. Großschädl, S. Tillich, C. Rechberger, M. Hofmann, and M. Medwed. Energy Evaluation of Software Implementations of Block Ciphers under Memory Constraints. In R. Lauwereins and J. Madsen, editors, *DATE*, pages 1110–1115. ACM, 2007.

[11] C. Herbst, E. Oswald, and S. Mangard. An AES Smart Card Implementation Resistant to Power Analysis Attacks. In *Applied Cryptography and Network Security, Second International Conference, ACNS 2006, volume 3989 of Lecture Notes in Computer Science*, pages 239–252. Springer, 2006.

[12] M. Joye, P. Paillier, and B. Schoenmakers. On Second-Order Differential Power Analysis. In Rao and Sunar [26], pages 293–308.

[13] M. Joye and J.-J. Quisquater, editors. *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*. Springer, 2004.

[14] B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2003.

[15] P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[16] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In Wiener [33], pages 388–397.

[17] S. Mangard. A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion. In P. J. Lee and C. H. Lim, editors, *ICISC*, volume 2587 of *Lecture Notes in Computer Science*, pages 343–358. Springer, 2002.

[18] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *WOST'99: Proceedings of the USENIX Workshop on Smartcard Technology*, pages 17–17, Berkeley, CA, USA, 1999. USENIX Association.

[19] I. Olkin and J. W. Pratt. Unbiased Estimation of Certain Correlation Coefficients. *Annals of Mathematical Statistics*, 29(1):201–211, 1958.

[20] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing (2nd ed.)*. Prentice-Hall, Inc., NJ, USA, 1999.

[21] S. B. Örs, F. Gürkaynak, E. Oswald, and B. Preneel. Power-Analysis Attack on an ASIC AES implementation. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2*, page 546, Washington, DC, USA, 2004. IEEE Computer Society.

[22] S. B. Örs, E. Oswald, and B. Preneel. Power-Analysis Attacks on an FPGA - First Experimental Results. In C. D. Walter, Çetin Kaya Koç, and C. Paar, editors, *CHES*, volume 2779 of *Lecture Notes in Computer Science*, pages 35–50. Springer, 2003.

[23] E. Peeters, F.-X. Standaert, and J.-J. Quisquater. Power and Electromagnetic Analysis: Improved Model, Consequences and Comparisons. *Integr. VLSI J.*, 40(1):52–60, 2007.

[24] E. Prouff, M. Rivain, and R. Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.

[25] J.-J. Quisquater and D. Samyde. ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards. In I. Attali and T. P. Jensen, editors, *E-smart*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer, 2001.

[26] J. R. Rao and B. Sunar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*. Springer, 2005.

[27] S. Chari and J. R. Rao and P. Rohatgi. Template Attacks. In Kaliski Jr. et al. [14], pages 13–28.

[28] S. Mangard and E. Oswald and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[29] F.-X. Standaert, S. B. Örs, and B. Preneel. Power Analysis of an FPGA: Implementation of Rijndael: Is Pipelining a DPA Countermeasure? In Joye and Quisquater [13], pages 30–44.

[30] F.-X. Standaert, S. B. Örs, J.-J. Quisquater, and B. Preneel. Power Analysis Attacks Against FPGA Implementations of the DES. In J. Becker, M. Platzner, and S. Vernalde, editors, *FPL*, volume 3203 of *Lecture Notes in Computer Science*, pages 84–94. Springer, 2004.

[31] S. Tillich and J. Großschädl. Power Analysis Resistant AES Implementation with Instruction Set Extensions. In P. Paillier and I. Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 303–319. Springer, 2007.

[32] S. Tillich and C. Herbst. Attacking State-of-the-Art Software Countermeasures-A Case Study for AES. In E. Oswald and P. Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 228–243. Springer, 2008.

[33] M. J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

[34] G. K. Yeap. *Practical Low Power Digital VLSI Design*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.