On the Efficiency of Design Time Evaluation of the Resistance to Power Attacks

Alessandro Barenghi[†], Guido Bertoni[‡], Fabrizio De Santis[*] and Filippo Melzani[‡]
[*]Institute for Security in Information Technology, Technische Universität München, 80333 Munich, Germany
Email: desantis@tum.de
[†]Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano (MI), Italy
Email: barenghi@elet.polimi.it
[‡]STMicroelectronics, 20041 Agrate Brianza (MB), Italy
Email: {guido.bertoni,filippo.melzani}@st.com

***Abstract*—Side-channel attacks are a realistic threat to the security of real world implementations of cryptographic algorithms. In order to evaluate the resistance of designs against power analysis attacks, power values obtained from circuit simulations in early design phases offer two distinct advantages: First, they offer fast feedback loops to designers, second the number of redesigns can be reduced. This work investigates the accuracy of design time power estimation tools in assessing the security level of a device against differential power attacks.**

I. Introduction

The need for ubiquitous security has lead to the use of cryptography in a large number of consumer grade devices such as PCs, smart phones, set-top boxes or smart cards. In the last decade, *side-channel attacks* have proven to be a threat far more effective than purely mathematical cryptanalytic attacks, proving that a motivated attacker can breach the security of standard cryptographic algorithms with a moderate economic effort [1]. Physical implementations of cryptographic algorithms allow an attacker to deduce pieces of information related to the secret key embedded in the device, through analyzing environmental parameters during the regular operation of the device. Common environmental parameters providing a side-channel include power-consumption, electromagnetic (EM) radiations and the timing needed to compute the results by the device. One of the most effective and relatively simple technique is represented by *power analysis*, which exploits the information leaked by the power-consumption of a device running a cryptographic primitive [2]. Since the dynamic power-consumption of a digital device is dependent on the switching-activity of logic gates, which in turn is correlated to the values being processed, the attacker may build a family of key-dependent models and check which one actually predicts correctly the real power-consumption of the circuit. This is done through predicting the switching-activity of a part of the circuit which is combining a known value (f.i. the plaintext) with a small portion of the key. The ability to model separately only a small part of the key is the main advantage of differential power analysis: the attacker is able to take into account small parts of the key independently, thus one needs to build only a small number of models.

Once all the switching-activity models for each possible value of the key portion considered are ready, the attacker proceeds to measure the actual power-consumption of the device via a digital oscilloscope. When actual measurements are available, the attacker evaluates the goodness of fit of the a-priori models to the actual distribution of the traces. Since the power-consumption grows linearly with the entity of the switching-activity, an effective measure of goodness of fit is represented by Pearson's linear correlation coefficient. Since the attacker does not know a-priori on which time instant the modeled computation is performed, the correlation analysis is performed time-wise for every instantaneous measure point of the power-consumption. The model built on the correct key hypothesis will show a significant correlation with the actual measurements when the modeled computation is performed, while all the other models will not. This enables the attacker to infer the correct value of the key portion [3].

This kind of security analysis requires the manufacturer to have a working prototype of the chip. This forces the security analyst to act as one of the last elements of the production chain, thus involving high re-engineering costs in case a product is found to be vulnerable to power attacks. A desirable objective is to perform the security assessment against power analysis at design time, but there are no such dedicated tools currently available. It is thus worth investigating the reliability of common power estimation tools when employed to generate accurate power-consumption traces to be used during power analysis attacks.

This work will present a comparative analysis of the security margins inferable from simulated and measured traces, in order to verify if the available tools for power estimation are effective when evaluating the security of digital devices at design time.

The remainder of the paper is organized as follows. Section II introduces the evaluation platform for our attacks, Section III describes the toolchain employed to obtain the simulated power traces and provides insights on the best practices discovered, Section IV gives an overview on the workbench employed to collect the measurements on the real world device and Section V presents the results

Research done while at STMicroelectronics.

of the comparison. Finally, Section VI summarizes our conclusions and points towards future research directions.

II. Target Implementation

This section describes the target hardware architecture and cryptographic algorithm employed as a practical benchmark.

A. Target Device

The target device of our evaluation is a development board from STMicroelectronics™ hosting an ultra low-power microprocessor geared towards healthcare applications, a typical scenario where security concerns are critical. The board features a 32-bit microprocessor fabricated with a 90nm cell library tuned for ultra low-power applications. The platform is equipped with a 66 kB on-die SRAM and an external 384 kB Flash memory and is clocked via a programmable PLL for system frequency generation, operating in the 4-48 MHz range. The regular operating frequency of the CPU clock is 4 MHz. Further details on the board model cannot be disclosed due to confidentiality issues.

B. Target Algorithm

The algorithm chosen for the evaluation is the Advanced Encryption Standard (AES): the symmetric key cipher chosen as the standard for secret key encryption by NIST [4]. The AES algorithm processes 128 bit of plaintext at once and encrypts them using a key that is 128, 192 or 256-bit wide depending on the security requirement of the application. Without loss of generality, we will be analyzing the 128 bit key version, as the same considerations can be successfully repeated for the higher security levels. The algorithm combines together the key material with the plaintext through the iteration of a round structure which comprises four basic operations. The inner state of the cipher is initialized with the 128 bits of the plaintext, from now on seen as a 4×4 byte state matrix, for the sake of clarity in description. The implementation in use employed a fully unrolled version of the algorithm, compiled with a `GCC` based toolchain targeted for the ultra low-power processor in use. The compiler was instructed not to perform optimizations (-O0) in order to obtain a clear scenario for analysis and comparison of attacks on simulated and measured traces. The first operation of the round is the SubBytes, a bytewise substitution of the state through a table lookup. The state is then bytewise rotated according to the pattern specified by the ShiftRows operation. Subsequently, the MixColumns combines together the values of the state columnwise, and is realized as a straightforward implementation of the FIPS-197 standard. Finally the AddRoundKey, an XOR addition of the key values to the actual state of the cipher, is performed 32 bit at a time in our platform.

III. Methodology for Simulated Power Traces at Gate-level

This section will provide an overview on the methodology we employed in order to generate simulated power-consumption traces of the target of our analysis. Simulated power traces are generated by automatic tools that estimate the power-consumption of a given digital circuit. The simulation can be performed at different levels of accuracy, depending on the needs of the designer. The most accurate way to predict the power-consumption for a given design is to perform a circuit simulation at transistor level, computing the value of the currents circulating for each time instant and obtain the instantaneous power-consumption value by multiplying them by the value of the supply voltage. However, the computational load imposed by this kind of simulation does not allow the designer to simulate large, real world circuits within reasonable time limits. In order to reduce the cost of the simulation, a common practice is to model the behavior of every logic gate present in a standard cell library only once. The gate is characterized in terms of its static power-consumption and dynamic (switching) power-consumption. All the models realized through transistor level simulation of the behavior of the single gates (or cells) of the chosen library are subsequently coalesced into a power library. Usually, the silicon provider provides the power libraries alongside the cell libraries available for printing to the designer, therefore the ASIC designer does not need to directly perform the power estimations for the cells. After obtaining the power library, the designer proceeds to simulate the gate level activity of the circuit only employing a netlist description of the design. Combining the information available from the power library with the switching-activity of the circuit obtained from the simulation the actual power profile of the device is built. In order to enhance the precision of the gate level power simulation, the tools are also able to exploit a description of the parasitic capacitances of the circuit net, representing the output and logic port loads, which are determined from the netlist by the synthesis tool. An overview of the workflow for obtaining simulated power traces at gate level is provided in Figure 1. In

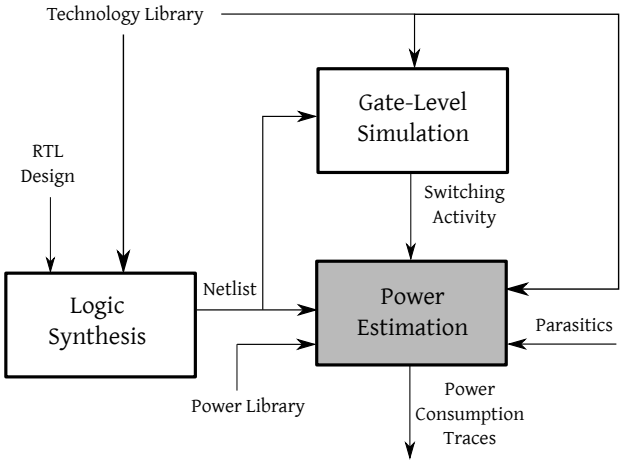


Figure 1. Workflow for Obtaining Simulated Power Traces at Gate Level

particular during the simulation steps, the nominal clock frequency f_{clk} must be provided to the gate-level simulator in order to inform the tool about the position of clock-edge events, together with the time step precision t_r at

which the simulation shall be run. In order to obtain a reliable simulation, f_{clk} must be lower or equal to the clock frequency provided as a target during the synthesis process to allow the switching-activity of the gates to terminate before the next clock cycle begins. On the other hand, the value of t_r, i.e. the simulation accuracy may be varied regardless of the actual clock frequency at which the circuit is designed to be working. Time resolution may take any value ranging from femtoseconds up to milliseconds, however the time needed for the simulation and the size of the output data sets are strongly influenced by this choice. In order to avoid the production of redundant information in the power traces, we analyzed the minimum gate delay, as specified by the Verilog `‘timescale` directive in the gate description within the library [5]. The value of t_r was set to the least value among the gate delays in order to perform a simulation as accurate as possible of the circuit under exam. Hence, the simulator proceeds to compute the power-consumption of the design through a finite events approach: every time step the dynamic power-consumptions related to the events which happened during the last time quantum are added together. This quantity is added to the static power-consumption of the device to obtain the total quantity of energy absorbed by the device during the simulation step.

IV. Methodology for Measured Power Traces

This section will provide a description of the experimental measurement setup employed to collect power traces from the physical circuit. In order to record measurements of the power-consumption of a circuit the prime instrument is a digital oscilloscope. Modern digital oscilloscopes are based on a general purpose computer endowed with a properly shielded board equipped with a fast, low noise Analog to Digital Converter (ADC). It is thus possible to record directly the sampled values on a mass storage memory and subsequently process them with a general purpose computer. The quality of measurements is influenced by the effective analog bandwidth provided by the sampling equipment circuits and the effective sampling rate of the ADC. A key choice is the one to employ an equipment accurate enough to capture the fastest changes in the dynamic power-consumption of the device under test. In this sense it is required that the analog bandwidth of the oscilloscope at least matches the working frequency of the circuit under exam, and that the sampling rate provided by the ADC allows to respect the Nyquist bound in order not to lose any information during the sampling. A partial analysis may be possible even with cheaper equipment, but the confidence of the security evaluation may be strongly affected. Another pair of key parameters are the resolution of the ADC, which directly impacts on the minimum value that can be sampled, and the noise floor of the instrument, which specifies the entity of the average noise introduced by the equipment. If the minimum value that can be sampled is lower than the differences in dynamic power-consumption that the attacker is able to model, the attack will not succeed even if the a-priori model is perfect. The other part of the equipment affecting significantly the measurements quality is the probe chosen to sense the changes in the power-consumption. Two methods of measuring the current flowing in a supply line are possible: either a current probe, based on Lenz's law effects is placed close to the desired line, or a voltage probe is employed to measure the variations in the voltage drops at the ends of a small resistor inserted in the supply line. Voltage probes are usually preferred in this measurement scenario due to the far higher bandwidth with respect to current probes, albeit their noise floor is higher than the current probes. The shunt resistor, can be inserted either between the device under test and the ground line or between the supply line at V_{DD} and the device. In case the second insertion point is chosen, the use of a differential probe is fundamental in order to reject the large common mode portion of the voltage present on the measurement point. A key requirement, for differential power analysis is the time alignment of the traces. The need for precise timing of the measurements is a result of the implicit assumption that the sensitive operation happens always in the same point in time. This in turn implies that, in order for the analysis to succeed, the measurements should be started by a signal step-locked to the beginning of the encryption. This can be achieved employing a second probe to sense a trigger signal. The trigger signal informs the oscilloscope to start recording samples for a predefined period of time fixed by the acquisition window $w = \frac{N}{f_s}$, where f_s if the sampling frequency and N the number of samples to store per trace. The trigger signal may either be an ad hoc assertion of a General Purpose Input/Output (GPIO) pin of the device, or the occurrence of the n-th clock cycle which can be detected by the oscilloscope.

V. Experimental Results

In this section we will now provide the technical details of our measurement and simulation setup, and proceed to compare the effectiveness of simulated power traces against measured ones in the evaluation of side-channel attacks. Measured power traces were obtained with an *Agilent Infiniium* 80000B series oscilloscope and an active Agilent differential voltage probe. The oscilloscope features 4 independent analog channels, a 2 GHz analog bandwidth, coupled with an 8-bit ADC capable of recording 40 GSamples/s, with a noise floor of 3 mV RMS, and a minimum vertical resolution of 10 mV. The measured power traces have been acquired using a sampling frequency of 50 MSa/s over an acquisition window of 2.6 ms. The sampling frequency is thus providing a fivefold margin over the Nyquist bound for sampling the 4 MHz working frequency of the chip. Figure 2 provides a schematic view of the measurement circuit. The measurements have been taken by sampling the voltage drop variations over the resistor R=1 $k\Omega$ inserted in series between the power pin of the chip and a low-drop voltage regulator (LDO) supplied by an external power supply. The role of the LDO is to stabilize supply voltage and shorten the path from the supply voltage to the measurement point in order to minimize the noise coming from the power line. The choice of a relatively big shunt resistor (commonly the value of

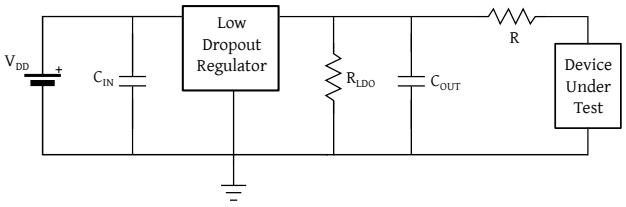


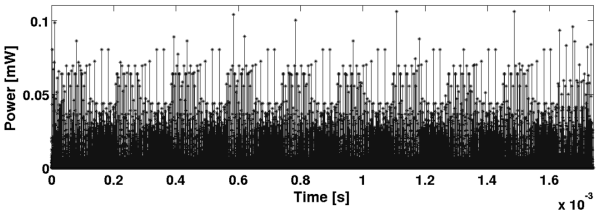
Figure 2. Schematic View of the Acquisition

the shunt resistor is in the Ω range) was driven by the amount of current (in the μA range) absorbed by the chip.

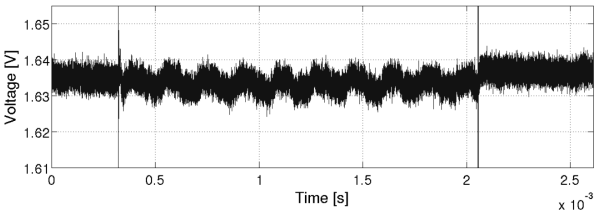
Simulated power traces have been obtained by estimating power-consumption of the microprocessor when running the target AES software implementation using standard tools used by semiconductor companies for synthesis, evaluation and validation of the digital designs. In particular, the first step to perform in order to obtain simulated power traces, i.e. the logic synthesis of the RTL design under attack, has been done using Encounter RTL Compiler from Cadence Design Systems. Subsequently, the gate-level simulation step has been done through instantiating the logic ports of the netlist and recording their switching-activity by stimulating the inputs of the design with a testbench file using Incisive Enterprise Simulator from Cadence Design Systems. Finally, once both the netlist level description of the circuit and its switching-activity during the computation of the AES cipher were obtained, the generation of the power traces step was performed using PrimeTime 2009 by Synopsys. To model a device matching the real one under exam, the chip has been synthesized using the same ultra low-power technology at 90 nm that has been used to fabricate the physical chip. It is worth noting that only the arithmetic-logic unit (ALU) and the interfaces to and from the SRAM holding the code and the data have been simulated in this case, in order to keep the space occupied by simulated traces and simulation time within practical feasibility. The storage and retrieval of the data from the SRAM was simulated through an HDL testbench file which provided the correct values on the signal wires going to the SRAM interface of the chip. The time resolution used for power estimation was 10 ns, thus resulting in simulated power traces capturing an effective bandwidth of 50 MHz. The choice of the time resolution was driven by the practical infeasibility of simulating the whole core employing as timescale the same as the minimum switching time of the circuit components. However, as the results in this section will confirm, the information loss from the attack evaluation standpoint was negligible, since the average switching-activity of the circuit are driven by a much slower clock front at 4 MHz.

A. Traces Analysis Comparison

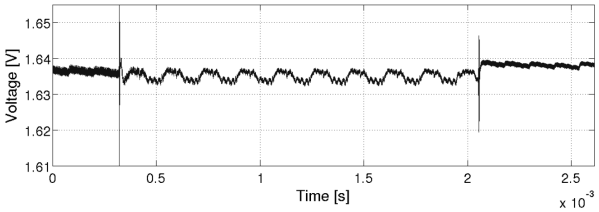
We will now proceed to present the results of the comparison on the obtained power traces. Figure 3 shows a comparison between the estimated and the measured power-consumption of a single AES encryption in the time domain. Precisely, Figure 3(a) provides the estimated power-consumption of one encryption starting at time 0, while Figure 3(b) provides the measured power-consumption of one encryption starting at time 323 ms. The beginning of the encryption operation on the measured traces can be spotted from the spike in power-consumption caused by the assertion of the trigger signal. In both cases, the activity of the ten AES rounds is clearly distinguishable in the traces and lasts in nearly 1.74 ms. However it is easy to notice, even upon a direct visual inspection that the measured trace is affected by a significant amount of noise. In order to reduce measurement and environmental noise, it is possible to collect a number of measurements of the same encryption trace and average them. The results of the averaging are depicted in Figure 3(c), showing a significant noise reduction. In order to



(a) Simulated Power Trace.



(b) Measured Power Trace.

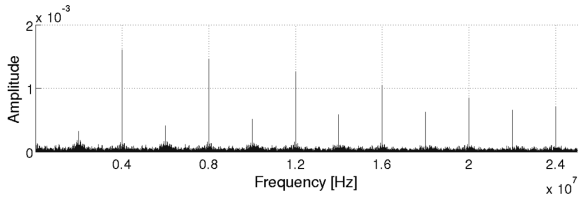


(c) Mean Measured Power Trace.

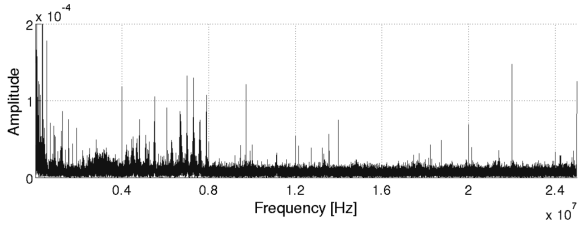
Figure 3. Trace Comparison in the Time Domain

characterize the informative content of the traces, Figure 4 shows a comparison between the estimated and the measured power-consumption of a single AES encryption in the frequency domain. The figures depict the single side amplitude spectrum of both measured signals and the simulated one. The first thing which can be noticed is that the harmonic component of the simulated power traces having the greatest amplitude is the one located at 4MHz, exactly the clock frequency of the device under exam. This is mutuated by the fact that the switching-activity, and thus the dynamic power-consumption, of a clocked circuit mostly happens in correspondence with the clock-edge [6].

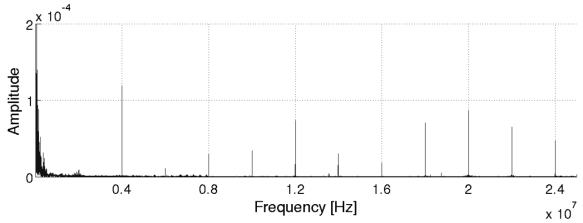
The simulated signal has a large effective bandwidth, since the last significant harmonic component showing a peak is located at 24 MHz. From the spectra plot it is even more evident how the single measurement is affected by a significant amount of noise, which is effectively reduced by averaging over multiple measurements. However, when comparing the spectrum of the averaged signal with the one of the simulation it is possible to notice that some of the harmonics are significantly damped. This is due to the RLC circuit that is established on the path from the measurement point to the ADC of the oscilloscope [7]. These parasitics are typically due to the capacitances that are intentionally placed by analog designers between V_{DD} and `GND` in order to stabilize the voltage supplied to the chip. Moreover, another effect to be taken into account is the one of the bonding wires connecting the pins of the package to the solder pads of the chip. This in practice results in the measured power traces being always to some extent band pass filtered when compared to simulated power traces. As a final analysis, we investigated the entity



(a) Single Simulated Trace.



(b) Single Measured Trace.



(c) Average Measured Traces.

Figure 4. Trace Comparison in the Frequency Domain

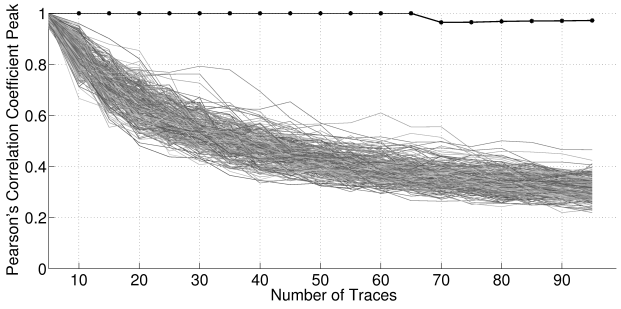
of the misalignment of the measured traces, since the simulated traces are perfectly aligned by construction. A straightforward comparison of the empirical distributions of the execution times of 100 different encryption has been performed. In this case, simulated and measured power traces have exactly the same execution times with an average of 1.731 ms and a sample standard deviation is 1.2 μs. If from one side, the same execution times provide an evidence for the soundness of acquisition setup, then from the other side the sample standard deviation highlights that encryptions do not always take the same amount of time. The maximum distance between the shortest and the longest execution is $\approx$ 16 clock cycles for our set of traces. Since there are no either mechanisms such as caches or instructions in the code such as conditional branches or comparisons that may affect the execution time, the cause of different duration has to be sought in the design of the instruction set. In fact, the decode stage of the CPU needs a different amount of time to fully decode some instructions, on the value of their actual operands. This phenomenon leads to a spread over time of the instants where the correct model will have non negligible correlation with the trace [8], while lowering the absolute value of the correlation coefficient for those points. In addition, the differences in the execution time depending on data represents another side-channel that can be exploited by an attacker for mounting timing attacks against the implementation [9].

B. Power Attacks Comparison

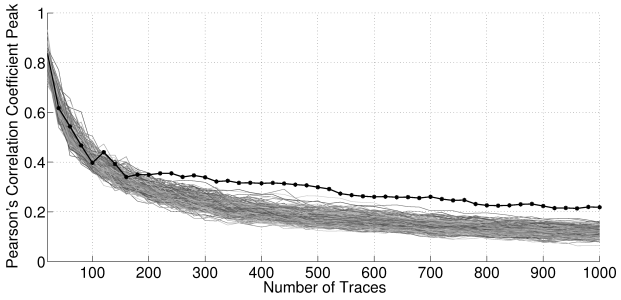
After providing an analysis on the differences between simulated and measured power traces, we will now present the effects induced by those differences on the performances of power attacks evaluation. As described in section I, the typical way to carry out correlation power attacks is to calculate the correlation coefficient between the hypothesized power-consumption of an encryption operation for every possible key guess and the actual power traces. In order to analyze the results, the values of the correlation coefficient for the different key hypotheses for each time instant are compared. The correlation for the correct key is expected to have some significant values in correspondence with the instants of time where the predicted intermediate values are actually employed by the circuit. The correlation analysis on simulated power traces has been carried out employing 100 power-consumption simulations of the circuit while 1000 measurements of the actual running circuit were collected. In order to suppress measurement noise, each of the 1000 measurements is obtained as the result of a sample-wise average of 32 encryptions with the same plaintext. The lower number of simulated power traces employed in the analysis is justified by the large amount of time and disk space needed to collect them. The second step in the evaluation of simulated traces was to evaluate the precision achievable with simulated power traces with another consumption hypothesis employed in the common literature to attack the AES cipher. In our case, this attack will target the power-consumption of the `load` operation involved to perform the lookup required to obtain the result of the `SubBytes` operation and all the instructions bound to the lookup (f.i. moving the value among registers or storing of the value in the target register). Figure 5 compares the results of an attack against simulated and measured power traces employing a single byte[1] of the state after the

[1]The pictures depict the results for the first byte of the state, as the other results are analogous

first `AddRoundKey` and the first `SubBytes` operations have been performed. Since both operations act bytewise, it is



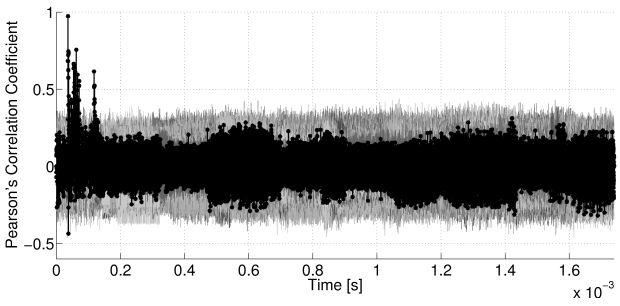
(a) Simulated Traces.



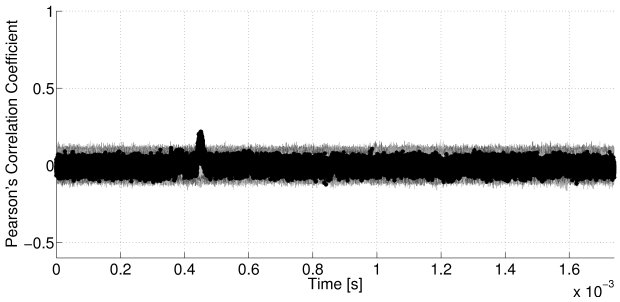
(b) Basic Traces.

Figure 5. Comparison of the Effectiveness of the Attack to the Output of the `SubBytes`

necessary to hypothesize a single byte of the key in order to build a consumption model. This results in building 2^8 consumption models employing as a power estimator the Hamming Weight (HW) of the result. This consumption model captures the power consumed by the circuitry, when holding the value, either during computation or load and store operations. The figure depicts the maximum correlation coefficient over a number of traces considered in the attack for every key hypothesis. The trend of the correlation coefficient of the model based on the correct key hypothesis is drawn in solid black, while the wrong ones are drawn in grey. It is worth noting that in the simulated case the attack reaches a correlation close to the theoretical maximum. This is due to the the traces being both noise free and perfectly synchronized, thus leading to an almost perfect linear correlation between the HW model and the simulated values. Instead, only a low peak of correlation (≈ 0.2) is reached with ten times more traces in the measured case. In this case, it is worth noting that using simulated power traces is possible to clearly distinguish the correct key from the wrong key guesses already with 20 traces, while using measured power traces ≈ 700 traces are needed to clearly distinguish the correct key from the wrong key guesses. This result can be explained by observing that simulated power traces are noise free while measured traces have a relative low SNR due to the practical difficulties to precisely measure the power absorbed by an ultra low-power device. After ascertaining the efficiency of employing simulated power traces to estimate the entity of power attacks, we proceeded to understand the timing accuracy of the correlation values. This analysis is particularly useful in order to precisely understand which part of the designed circuit is leaking information while processing the sensitive values. Figure 6 compares the results of an attack conducted with the same modelling and hypotheses mentioned before. The results are shown for 100 simulated traces and 1000 measured ones. The results show that there is only one single



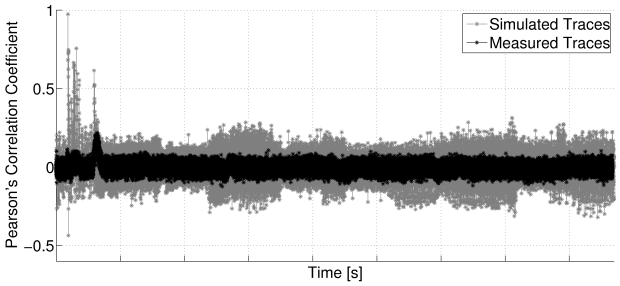
(a) Simulated Traces



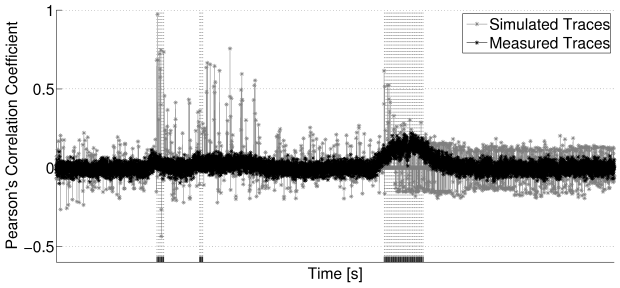
(b) Measured Traces with Basic Measurement Board

Figure 6. Power attack to the Output of the `SubBytes` Operation in the First Round

time instant where the correlation spikes in the measured traces, while there are some peaks in the simulated traces in this case. This fact suggests that either there may exist some operations that leaks more than others (i.e. provides a larger SNR) or that the intrinsic impedance of the measured circuit filters away some contributions in time. In order to better understand the relationship between the two results, Figure 7 shows a superposition of simulated and measured traces for the second attack considered in this section. In order to provide a fair comparison, the traces have been realigned in time to the beginning of the AES encryption. Only the traces for the correct key are shown in this case for the sake of clarity. As it can be noticed, the lone peak in the correlation of the actual measurements matches perfectly in time one of the peaks predicted by the simulation. Willing to understand the multiple peaks on the simulated traces, we analyzed the code being executed on the CPU (provided in Listing 2). All the time intervals in which the attacked value is in use are highlighted by the dashed zones in Figure 7. It is worth noting that the byte under attack is used by several different instructions in the code and every time

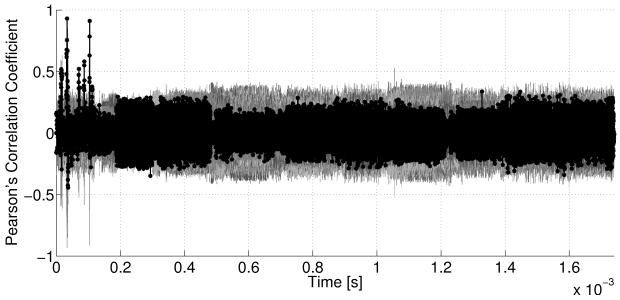


(a) Normal View.

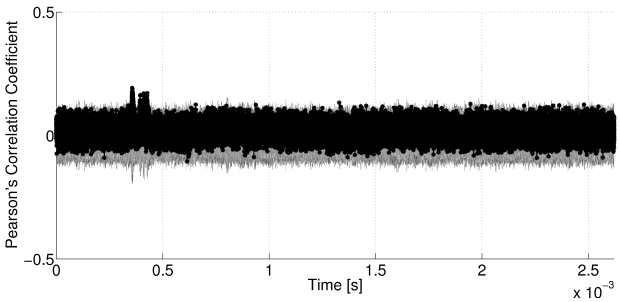


(b) Zoomed View.

Figure 7. Superposition of Attacks to the Output of the `SubBytes` Operation



(a) Simulated Traces.



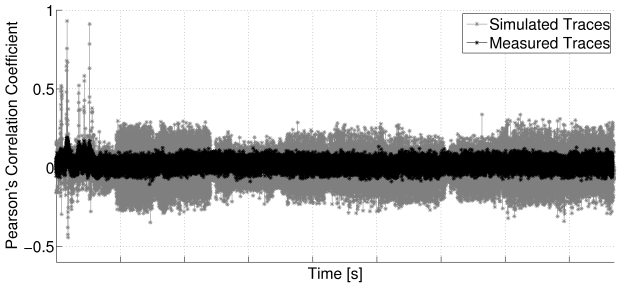
(b) Measured Traces with Basic Measurement Board.

Figure 8. Power attack to the Output of the first `AddRoundKey` Operation

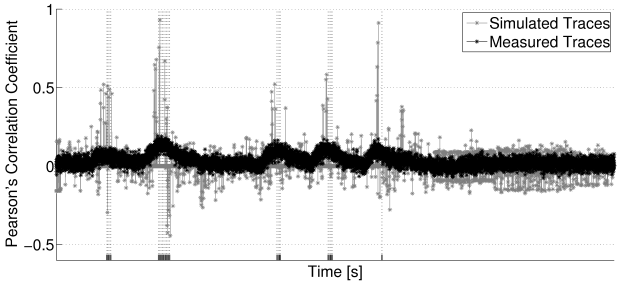
it generates a peak of correlation in the simulated power traces, but not in the measured power traces where the peak of correlation is well visible only where there is an high density of `load` operations. We reckon that the `load` operation consumes a greater overall amount of energy, thus the differences in consumption are easily measurable even with the technical difficulties imposed by the real world setup.

Willing to investigate also the ability to predict the effects of a computational operation, we evaluated the results of an attack to a single byte of the output of the first `AddRoundKey` operation. Similarly to the previous scenario the key hypothesis made is on a single byte of the key value and the consumption model considered is the Hamming Weight of the output of the operation. Figure 8 depicts the correlation coefficients over time obtained from both the simulated traces and the measured ones. In the simulated traces, it is clearly evident, with a correlation value very close to the maximum for the correct key hypothesis model, where the circuit is performing the sensitive operations. By contrast, some wrong key hypotheses are negatively correlated with the actual power-consumption. The reason is to be sought in the intrinsic symmetry of the consumption model chosen, which will yield together with a correct hypothesis, also a completely wrong one. In the measured traces, the instants in time where the operations take place are still evident, although a much lower correlation coefficient results from the analysis. The reason for correlation coefficient plots showing significant correlation in more than a single instant in time is the fact that byte under estimation is manipulated by the CPU in more than a single instruction as shown in Listing 1.

Similarly to the previous analysis, we proceeded to compare the time accuracy of the prediction with the one of the measured traces. Figure 9 shows a superposition of the correlation values of simulated and measured traces. From Figure 9, it can be easily seen that the peaks in both



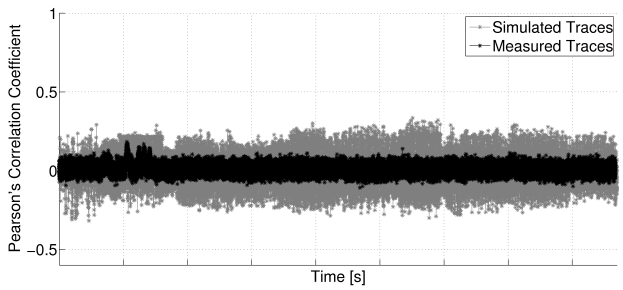
(a) Normal View.



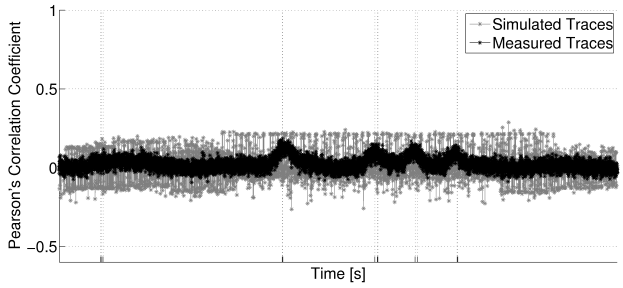
(b) Zoomed View.

Figure 9. Superposition of Attacks to the Output of the first `AddRoundKey` Operation

simulated and measured traces match the expected timing where the operations involving the attacked byte are performed. In particular, in the simulated trace the highest peaks are in an exact correspondence with the instants in which only the byte under attack is used, while lower peaks are in correspondence with instants where the attacked byte is manipulated together with some other bytes (f.i. when other values in the same register are modified and the byte is saved again without modification). However, the correlation peaks in the measured traces are only present in correspondence of the memory `load` and `store` operations, while there is no significant correlation in the processing instructions. This fact provides further evidence that there are some operations which are more prone to information leakage and thus must be protected with greater attention, while confirming the timing accuracy of the simulation based analysis. Finally, willing to investigate the effects of the architectural operation taking different time to be computed depending on the operands, we performed a last analysis. This time we targeted one byte of the output of the first round of the AES cipher. Since our interest is to check the precision of the predictions for the correct key, this time, instead of performing an attack, we computed the correlations for the correct key hypothesis (which encompasses 32 bits). Figure 10 provides a superposition of simulated and measured traces for an attack to the output of the first round. As it can be seen,



(a) Normal View.



(b) Zoomed View.

Figure 10. Superposition of Attacks to Output of the First Round

the time shifts caused by the different timings required to decode some instructions significantly lower the correlation values in the simulated traces. In particular, since the only instruction which had never been present in the code of the examined AES round primitives is the `mul` operation (as reported in Listing 3), we deem that operation to be one requiring different times for decoding. A noteworthy side effect is that the `mul` instruction may thus represents a potential source of leakage that can be exploited by timing attacks. It is worth noting that small peaks of correlation show up in the measured traces, while for simulated traces the correlation is spread without really emerging like a distinguishable peak. The explanation has to be sought in the number of traces considered in the two attacks, which lowers the confidence interval for the estimate of Pearson's correlation coefficient.

VI. Conclusion

In this work we proposed the results of the evaluation of the efficiency of common power estimation tools when employed to predict the vulnerability of secure chips to side-channel attacks at design time. The results obtained have shown that differential power attacks can be successfully lead against simulated power traces. The time instants in which the simulated power traces show detectable correlation with the power model employed during the attacks match the expected timing for the sensitive operations. Moreover, the timing is also coherent with the results from actual traces measured from the real chip, while providing a greater accuracy. Therefore, the match between simulated and measured attacks allows to conclude that current tools for power estimation offer an interesting option for evaluating the security of cryptographic devices against differential power analysis attacks at design time.

In particular, it has been shown that the only difference between attacks against simulated and measured power traces lays in the effectiveness, that in the case of measured traces is typically worse due to a small SNR in the measurements also due to the difficulties to sample the power-consumption of an ultra low-power device and the inability of sampling the high frequencies in practice. The advantage of performing power analysis against simulated traces is that the source of leakage can be easily identified, isolated and studied individually. In particular, in our case study, it has been shown that there exist some instructions (precisely, `load` and `store`) that leak more information than others and therefore particular care should be taken to protect them. By contrast, implementations that avoid the use of such instructions should be considered more secure. Moreover, the analysis of simulated traces has also shown that there exists one instruction (precisely, `mul`) that can lead to timing attacks and thus particular care for security should be paid also at digital level.

The workflow presented here can be reused to assess the security of other different implementation of AES, eventually using different optimization levels or implementing different countermeasures against differential power attacks.

Acknowledgment

The authors would like to thank their collegues at STMicroelectronics, Sara Bocchio and Marco De Fazio, for the technical support provided in simulating and acquiring power traces, respectively.

References

[1] S. Mangard, M. E. Oswald, and T. Popp, *Power Analysis Attacks - Revealing the Secrets of Smart Cards*, 2007.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," *Lecture Notes in Computer Science*, vol. 1666, pp. 388–397, 1999.

[3] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, ser. Lecture Notes in Computer Science, M. Joye and J.-J. Quisquater, Eds. Springer Berlin / Heidelberg, 2004, vol. 3156, pp. 135–152.

[4] "Specification for the advanced encryption standard (aes)," Federal Information Processing Standards Publication 197, 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[5] V. Sagdeo, *The Complete VERILOG Book*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[6] A. Barenghi, G. Pelosi, and Y. Teglia, "Improving First Order Differential Power Attacks Through Digital Signal Processing," in *SIN*, 2010, pp. 19–29.

[7] H. Li, "Security evaluation at design time for cryptographic hardware," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-665, Apr. 2006. [Online]. Available: http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-665.pdf

[8] S. Mangard, "Hardware countermeasures against dpa - a statistical analysis of their effectiveness," in *Topics in Cryptology - CT-RSA 2004*, ser. Lecture Notes in Computer Science, T. Okamoto, Ed. Springer Berlin / Heidelberg, 2004, vol. 2964, pp. 1998–1998.

[9] D. J. Bernstein, "Cache-timing attacks on AES," 2004. [Online]. Available: http://cr.yp.to/antiforgery/cachetiming-20050414.pdf

[10] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002.

Appendix

The code below performs the operations for a generic round in the following order:

- 4 `AddRoundKey` operations in a row considering 4 bytes of the state at a time.
- 16 `SubBytes` and `ShiftRows` operations for each byte of the state.
- If the current round is not the last one, then the `MixColumns` transformation is performed using a straightforward implementation of the `xtimes` operation [10].

```
ldp %r0, -4[%r4]
ldw %r1, [%r0]  ; load plaintext
ldp %r2, -16[%r4]
ldw %r0, [%r2]  ; load key
movw %r2, %r1
xorw %r2, %r0  ; first AddRoundKey
stw -36[%r4], %r2
:
ldw %r0, -36[%r4]
srlw %r0, 16
srlw %r0, 8
movp %r1, %r0
```

Listing 1. first `AddRoundKey`

```
movp %r1, %r0
movp %r0, Sbox
addp %r0, %r1
ldb %r0, [%r0]  ; S-Box LUT
uextb %r0, %r0
movw %r2, %r0
slaw %r2, 16
slaw %r2, 8
:
orw %r2, %r0
:
orw %r2, %r0
:
movw %r1, %r2
orw %r1, %r0
stw -52[%r4], %r1
:
ldw %r0, -52[%r4]
:
ldw %r0, -52[%r4]
:
ldw %r0, -52[%r4]
:
ldw %r0, -52[%r4]
:
ldw %r1, -52[%r4]
:
ldw %r0, -52[%r4]
:
ldw %r0, -52[%r4]
```

Listing 2. `SubBytes` in the First Round

```
ldw %r0, -52[%r4]
andw %r0, #2139062143
addw %r0, %r0
movw %r1, %r0
ldw %r0, -52[%r4]
andw %r0, #-2139062144
srlw %r0, 7
mulw %r0, #27  ; mul operation
movw %r2, %r1
xorw %r2, %r0
ldw %r0, -52[%r4]
andw %r0, #2139062143
addw %r0, %r0
movw %r1, %r0
ldw %r0, -52[%r4]
andw %r0, #-2139062144
srlw %r0, 7
mulw %r0, #27  ; mul operation
xorw %r0, %r1
ldw %r1, -52[%r4]
xorw %r0, %r1
rotw %r0, 7
rotw %r0, 7
rotw %r0, 7
rotw %r0, 3
movw %r1, %r2
xorw %r1, %r0
ldw %r0, -52[%r4]
rotw %r0, 7
rotw %r0, 7
rotw %r0, 2
xorw %r1, %r0
ldw %r0, -52[%r4]
rotw %r0, 7
rotw %r0, 1
xorw %r1, %r0
ldp %r2, -16[%r4]
ldw %r0, [%r2]
movw %r2, %r1
xorw %r2, %r0
stw -36[%r4], %r2
```

Listing 3. `MixColumns` in the First Round